AD-783 804

TERCOM PERFORMANCE: ANALYSIS AND SIMULATION

Mark W. Cannon, Jr.

Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, Ohio

June 1974

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Organizations and individuals receiving announcements or reports via the Aerospace Medical Research Laboratory automatic mailing lists should submit the addressograph plate stamp on the report envelope or refer to the code number when corresponding about change of address or cancellation.

Do not return this copy. Retain or destroy.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22151

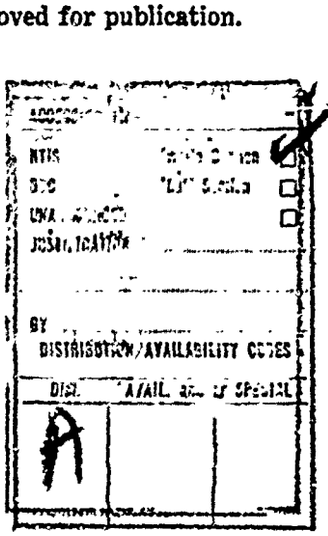
This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DODD 5230.0. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

HENNING E. VON GIERKE
Director
Biodynamics and Bionics Division
Aerospace Medical Research Laboratory

AIR FORCE/56780/17 July 1974 — 100



AD 783 804

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AMRL-TR-73-130	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: TERCOM PERFORMANCE: ANALYSIS AND SIMULATION	5. TYPE OF REPORT & PERIOD COVERED: Final Report - 19 Oct 72 to 15 Nov 73
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mark W. Cannon, Jr. Joseph W. Carl	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: 6570 Aerospace Medical Research Laboratory (BBM) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: PROGRAM ELEMENT: 62202F PROJ.: 7233; TASK: 7233-05 WORK UNIT: 7233-05-17
11. CONTROLLING OFFICE NAME AND ADDRESS: 6570 Aerospace Medical Research Laboratory Aerospace Medical Div, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433	12. REPORT DATE: June 1974 13. NUMBER OF PAGES: 71 ~~67~~
14. MONITORING AGENCY NAME & ADDRESS *(If different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

15. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19 KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

TERCOM
Navigation
Terrain Contour Matching
Pattern Recognition
Nearest-Prototype Classifier

20 ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A theoretical analysis of the performance of a nearest-prototype classifier (nearest in the sense of Euclidean distance) is presented as an approximate analysis of the performance of the Terrain Contour Matching Navigation System (TERCOM). The analysis is used to predict classifier performance on both artificially generated terrain with Gauss-Markov statistics and on real-world terrain.

(Continued on reverse side.)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

71

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

BLOCK 20. ABSTRACT (cont)

Over 150 simulation runs were performed over such terrain having added Gauss-Markov noise (independent of the terrain process) with 6 different noise levels. Results of the simulation are presented, giving performance as a function of signal/noise ratio and as a function of terrain roughness. Measured miss-distances are also presented. The simulation also includes the performance of a nearest-prototype classifier based on mean absolute distance (the first Minkowski metric), which is the actual metric employed in TERCOM.

The following conclusions are reached: (1) the analysis offers a useful description of actual TERCOM performance on both artificial and real terrain; (2) the current model is different from, and more convenient to use than the model previously available; (3) the TERCOM classifier should be slightly modified with respect to the mean-removal technique in order to significantly improve the theoretical and the actual performance; (4) TERCOM is an effective navigation system provided that the terrain possesses certain statistical characteristics; and (5) the necessary statistical characteristics of the terrain can be summarized in two parameter ratios, σ_Z/σ_T and σ_Z/σ_N, which relate to terrain and noise variance, and which are defined in the report.

SUMMARY

PROBLEM

An analysis of the Terrain Contour Matching Navigation System (TERCOM) was performed by the system's manufacturer, LTV E-Systems, Inc. An independent evaluation of that analysis and of TERCOM was desired for several reasons: (1) some doubt existed about initial assumptions; (2) that analysis was performed for a mean-square-distance (MSD) classifier with continuous input, but a mean-absolute-distance (MAD) classifier with discrete input is used in the actual hardware; and (3) the utility of the single terrain parameter, σ_z, suggested by E-Systems as being predictive of TERCOM performance seemed limited because it varies widely over terrain areas of interest.

APPROACH

The approach was twofold:

(1) Starting with basic assumptions, derive an error model for TERCOM. Concentrate on the MSD classifier because of its mathematical tractability, but treat the MAD classifier also as far as possible.

(2) Determine by computer simulation the performance capabilities of MSD and MAD, using both artificial Gaussian terrain and real terrain samples. Compare TERCOM performance over artificial statistically controlled terrain with the performance over real terrain to yield the parameters necessary for a predictive performance model.

RESULTS

A mathematical model for TERCOM "false-fix" probability was derived that differs from the original LTV E-Systems model in certain assumptions. False fix probability is presented as a finite sum having parameters that depend on sample size, terrain and noise correlations, and the signal-to-noise ratio.

The computer simulation revealed that a dramatic increase in the probability of a correct fix could be achieved by a simple change in the normalization procedure for the MSD or MAD computations.

It was found that the simulated performance of TERCOM on real terrain was the same as on artificial terrain, generated from a Gaussian process, when a parameter ratio σ_z/σ_t was the same for both terrains. This performance was specified by plots of probability of correct fix vs the parameter ratio σ_z/σ_n. A family of performance curves was generated by varying the parameter ratio σ_z/σ_t.

The error model developed herein is a good predictor of simulation results, but a rather poor predictor of performance on actual terrain. This is believed to be principally due to the assumption of stationary statistics for the terrain. While a Gauss-Markov model may be valid for local terrain regions, the Gauss-Markov statistics vary widely over real terrain, even over physical extents represented by the reference arrays encountered in TERCOM. Both the E-Systems error model and the present one offer insight into TERCOM performance, and both would undoubtedly be improved by considering the nonstationarity of the terrain statistics. The present model is not as computationally involved as the E-Systems model, however, and may therefore be preferable.

CONCLUSIONS

TERCOM performance can be significantly improved by means of a minor modification in the mean-removal procedure.

Present error models of TERCOM suffer from an assumption of stationary statistics. This assumption is not justified on real terrain except when precautions are taken to insure its validity.

A parametric family of performance curves can be generated by simulation of TERCOM flights over modified Gaussian terrain. Once such a family of curves is generated for a particular terrain sample size, all performance characteristics of interest are determined. It still remains to compare these results with actual flight test data.

TERCOM is an effective system provided that reference arrays can be obtained that produce acceptable performance. Such reference arrays are characterized herein, but their availability in the real world was not examined.

PREFACE

The work discussed in this report was performed under Project 7233, "Applications of Biological Principles as Solutions to Air Force Needs in Signal Processing and Information Handling," Task 7233-05, "Automatic Pattern Recognition for Air Force Systems," Work Unit 7233-05-17, "Application of Visual Pattern Recognition to USAF Problems." The analysis and simulation were performed in the Mathematics and Analysis Branch, Biodynamics and Bionics Division, 6570 Aerospace Medical Research Laboratory, during the period 19 October 1972 to 15 November 1973. The work was performed at the request of the Subsonic Cruise Armed Decoy (SCAD) System Program Office (ASD/RW86).

The authors thank Dr. O.H. Tallman, III, Col, USAF, and Dr. H.L. Oestreicher for useful discussions and encouragement during this effort. Thanks are also due Capt J. Gobien who suggested use of the Toeplitz approximation in the mathematical analysis, and Capt D. Brungart who provided the computer graphics.

INTRODUCTION

The terrain contour matching navigation system (TERCOM) (Ref 1) has been proposed as a self-contained, autonomous system for updating an inertial guidance system at selected checkpoints en route. The system is based on a pattern classifier that may be termed a nearest-prototype classifier, where "nearest" is defined in the first-Minkowski-metric sense. In order to obtain a clearer understanding of the capabilities and the limitations of the system, an analysis and evaluation of TERCOM was performed by this laboratory at the request of the monitoring agency.

The report of that analysis and evaluation is organized as follows. The first section presents a brief discussion of the performance of an error model for TERCOM that was developed by TERCOM's manufacturer. Some theoretical results are presented as suggestions for improving the model. A brief comparison of nearest prototype classifiers based on the first Minkowski metric (termed MAD classifiers) and based on the Euclidean metric (termed MSD classifiers) is also included. The next section presents a discussion of the simulation procedure and the simulation results. The last section presents the conclusions and recommendations.

THEORETICAL CONSIDERATIONS

PERFORMANCE OF THE TERCOM ERROR MODEL

Because the theoretical aspects of pattern recognition in general, and TERCOM system performance in particular, are quite mathematical, there must be some fairly strong motivation for the typical systems designer to consider them in depth. The initial motivation for development of an error model is, of course, to predict systems performance: such predictions are essential to the systems buyer and user. But, if the TERCOM manufacturer has provided an error model, why do we need more mathematical, theoretical considerations?

The answer is obtained by constructing a scatter plot of the predictions of the manufacturer's error model against actual flight test data.

Such a plot would reveal that predictions do not correspond well with the measured performance. The Pearson correlation coefficient computed for such a scatter plot ($\rho = 0.617$) is interpretable as meaning that the error model accounts for only about 38% of the variance in the measured performance. This leaves about 62% of the variance unaccounted for. When we consider the cost of a system like TERCOM, there is clearly strong motivation to reconsider a mathematical error model.

GENERAL CONCEPTS

Let $\bar{x} = (x_o, x_1, \ldots, x_{d-1})$ be the true-position, stored, sampled terrain contour. Assume $\bar{x}$ is accurate to some "mapping" error that is negligible compared to the observation noise. Let $\bar{r} = (r_o, r_1, \ldots, r_{d-1}) = \bar{x} + \bar{n}$, where $\bar{n} = (n_o, n_1, \ldots, n_{d-1})$ is the observation noise. Let $\bar{y}(\tau) = (y_o, y_1, \ldots, y_{d-1})$ be the stored sampled terrain contour at a geographical distance τ from the true position, as shown in Figure 1. Finally, let m denote the match-distance (e.g., the decision statistic) between the vectors $\bar{r}$ and $\bar{y}(\tau)$, and let m_o denote the special case $\tau = 0$.

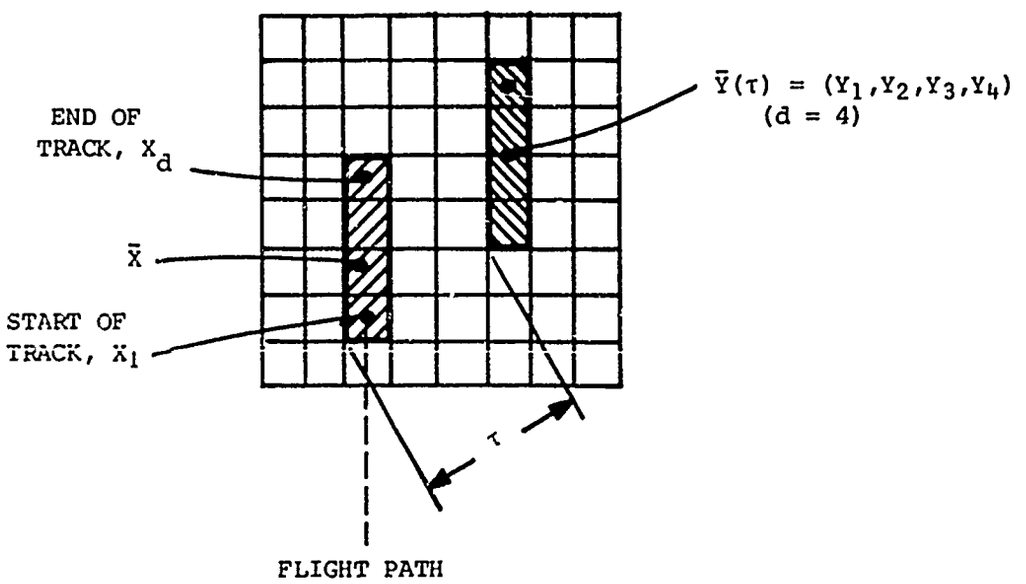


Figure 1. Nomenclature and notation for vector samples from a stored reference array.

First of all we indicate the procedure for finding the probability that $m < m_0$ for some fixed τ. This will show us what distributions we need to find in order to arrive at numerical predictions.

We let $p_\tau(m,m_0)$ denote the joint probability density function of m and m_0. Recall that m corresponds to some τ, and the function depends (in general) on this τ: hence the subscript. Then we compute

$$\text{Prob } \{m < m_0 | m_0\} = \int_0^{m_0} p_\tau(m,m_0)\ dm$$

When we allow m_0 to range over all permissible values we arrive at

$$\text{Prob } \{m < m_0\} = \int_0^{\infty} \int_0^{m_0} p_\tau(m,m_0)\ dm\ dm_0$$

We now make our first assumption. We assume that for sufficiently large τ, the distributions of m and m_0 are independent. In the case of the Gauss-Markov model, which we introduce shortly, this corresponds to $\tau > L_T$, where L_T is the correlation length of the terrain. This is equivalent to the following assumption:

$$\text{Prob } \{m < m_0\} = \int_0^{\infty} p(m_0) \int_0^{m_0} p_\tau(m)\ dm\ dm_0$$

Now we want to know the probability that m_0 is smaller than m for every permissible value of τ. Suppose that $p_\tau(m)$ is the same for each of the permissible values of τ (but, recall, $\tau > L_T$), and that the integral above has been evaluated to, say, β. That is,

$$\text{Prob } \{m < m_0\} \triangleq \beta$$

Then $$\text{Prob } \{m_0 < m\} = 1 - \beta \triangleq \alpha$$

Note that supposing $p_\tau(m)$ does not change with τ imposes a second assumption: that the terrain statistics are stationary.

To proceed, consider the rank ordering of three observations that correspond to m_0, m_1, and m_2, where m_1 and m_2 are match values for two distinct $\tau > L_T$. Six orderings are possible; suppose the probability of the i^{th} order is p_i.

m_0	m_1	m_2	p_1
m_0	m_2	m_1	p_2
m_1	m_0	m_2	p_3
m_1	m_2	m_0	p_4
m_2	m_0	m_1	p_5
m_2	m_1	m_0	p_6

We note that Prob $\{m_0 < m_1\} = p_3 + p_4 + p_6 = \alpha$, and Prob $\{m_0 < m_2\} = p_4 + p_5 + p_6 = \alpha$. It is a simple matter to compute Prob $\{m_0 < m_1 \bigcap m_0 < m_2\} = p_4 + p_6$. We further note the constraint $\sum_{i=1}^{6} p_i = 1$. These facts together imply that

$$\text{Prob}\{m_0 < m_1 \bigcap m_0 < m_2\} = 2\alpha - 1 + p_1 + p_2$$

Now, as α approaches one, it is clear that p_1 and p_2 must approach zero. Hence, for "large" α,

$$\text{Prob}\{m_0 < m_1 \bigcap m_0 < m_2\} \tilde{=} 2\alpha - 1$$

As α approaches zero, $p_1 + p_2$ dominate. If we suppose $p_1 \tilde{=} p_2 \tilde{=} 1/2$, we obtain

$$\text{Prob}\{m_0 < m_1 \bigcap m_0 < m_2\} \tilde{=} \alpha/2$$

The somewhat surprising thing is that this sort of combinatorial argument can be generalized to rank orderings of many observations: the approximations remain the same.

The approximations just developed are fine for the high and low ranges of α. But what about $\alpha \tilde{=} 1/2$? One case where α equals 1/2 arises when $p_i = 1/N$ for all i, and N is the number of possible rank orderings.

If M is the number of observations being ranked, then N = M! and Prob {m_0 is the minimum observation} = 1/M. We see that in this particular case, the approximation for "large" α gives the best result when M is large.

Using the preceding combinatorial argument as motivation, we choose the following approximation.

$$\begin{aligned}\text{Prob } \{m_0 < \text{all } m\} &\cong \max \{2\alpha-1, 0\} \\ &= \max \{1-2\beta, 0\} \\ &= \max \{1-2 \int_0^\infty p(m_0) \int_0^{m_0} p_\tau(m)\ dm\ dm_0, 0\} \end{aligned} \tag{1}$$

This expression approximates the probability of correctly assigning $\bar{r}$ to a position that is within a distance of L_T of $\bar{x}$.

At this point we see what distributions are of principal interest: $p(m_0)$ and $p_\tau(m)$. We must consider the definition of the classifier's match function before we can proceed further.

THE GAUSS-MARKOV STATISTICAL MODEL

The statistics of the terrain process and the noise process are assumed to be Gauss-Markov, and independent of each other. Hence the probability density function (using the notation of Ref 5)

$$p_{\bar{x}}(\bar{\alpha}) = (2\pi)^{-d/2} |\Lambda_T|^{-1/2} \exp \{-\frac{1}{2} \bar{\alpha} \Lambda_T^{-1} \bar{\alpha}^T\}$$

governs the terrain process, and the probability density function

$$p_{\bar{n}}(\bar{\alpha}) = (2\pi)^{-d/2} |\Lambda_n|^{-1/2} \exp \{-\frac{1}{2} \bar{\alpha} \Lambda_n^{-1} \bar{\alpha}^T\}$$

governs the noise process. The covariance matrix for the terrain, Λ_T, has i-jth elements given by

$$\sigma_T^2\ e^{-|i-j|/L_T}$$

The covariance matrix for the noise, Λ_n, has i-jth elements of the same form, but with different parameters,

$$\sigma_n^2 \, e^{-|i-j|/L_n}$$

The parameters L_T and L_n are called the correlation lengths of the terrain and the noise, respectively.

The joint distribution of $\bar{x}$ and $\bar{y}(\tau)$ has a similar form, but the covariance matrix is slightly more complicated. Define

$$\Lambda_z = \left[\begin{array}{c|c} \Lambda_T & \Lambda_{xy} \\ \hline \Lambda_{xy} & \Lambda_T \end{array}\right]$$

where Λ_T is the terrain covariance matrix, as before, and Λ_{xy} contains the covariance information between $\bar{x}$ and $\bar{y}(\tau)$. If we let the vector $\bar{x}$ be comprised of the i-jth cells (e.g., j is fixed and i runs from I through I+d; d is the dimension of $\bar{x}$) of the reference array, and let $\bar{y}(\tau)$ be the k-lth cells, then Λ_{xy} has elements given by

$$\sigma_T^2 \, e^{-\sqrt{(i-k)^2 + (j-l)^2}/L_T}$$

We also know the main diagonal terms are given by

$$\sigma_T^2 \, e^{-\tau/L_T}$$

Now we let $\bar{z} = [\bar{x} \,|\, \bar{y}(\tau)]$ be the 2d-dimensional vector formed by concatenating $\bar{x}$ and $\bar{y}(\tau)$. Then the joint distribution of $\bar{x}$ and $\bar{y}(\tau)$ is given by

$$p_{\bar{x},\bar{y}}(\bar{\alpha}_1,\bar{\alpha}_2) = p_{\bar{z}}(\bar{\alpha}) = (2\pi)^{-d}|\Lambda_z|^{-1/2} \exp\{-\frac{1}{2}\bar{\alpha}\Lambda_z^{-1}\bar{\alpha}^T\}$$

where $\bar{\alpha}_1$ and $\bar{\alpha}_2$ are the "halves" of $\bar{\alpha}$ corresponding to values of $\bar{x}$ and $\bar{y}(\tau)$. In the special case that τ is large enough Λ_{xy} is essentially a null matrix, and Λ_z has a block-diagonal structure. Then

$$p_{\bar{z}}(\bar{\alpha}) = p_{\bar{x}}(\bar{\alpha}_1)p_{\bar{x}}(\bar{\alpha}_2)$$

THE DISTRIBUTION ON THE DIFFERENCE VECTOR

The match functions defined by both the MSD and the MAD classifiers involve operations on the components of a difference vector. We now want to compute this difference vector and determine its underlying probability density function.

When TERCOM samples terrain it is, of course, sampling the true fix point, $\bar{x}$. This observation is perturbed by additive independent noise. The observation is, therefore

$$\bar{r} = \bar{x} + \bar{n}$$

Since $\bar{x}$ and $\bar{n}$ are independent, the distribution on $\bar{r}$ is easily computed (e.g., via characteristic functions) to be

$$p_{\bar{r}}(\bar{\alpha}) = (2\pi)^{-d/2} |\Lambda_r|^{-1/2} \exp\{-\frac{1}{2}\bar{\alpha}\Lambda_r^{-1}\bar{\alpha}^T\}$$

where $$\Lambda_r \triangleq \Lambda_T + \Lambda_N$$

The difference vector that the classifier uses is then

$$\bar{d} \triangleq \bar{r} - \bar{y}(\tau)$$

In the special case that $\bar{x}$ and $\bar{y}(\tau)$ are uncorrelated (e.g., large enough τ) the distribution on $\bar{d}$ is as easy to obtain as the distribution on $\bar{r}$.

$$p_{\bar{d}}(\bar{\alpha}) = (2\pi)^{-d/2} |\Lambda_d|^{-1/2} \exp\{-\frac{1}{2}\bar{\alpha}\Lambda_d^{-1}\bar{\alpha}^T\}$$

where $$\Lambda_d = \Lambda_r + \Lambda_T = 2\Lambda_T + \Lambda_n.$$

The more general case of correlated $\bar{x}$ and $\bar{y}(\tau)$ is harder to analyze, and is not considered herein.

THE MSD CLASSIFIER

In this section we want to compute the statistics on the match function for the MSD classifier, and then compute the approximate probability of correct assignment according to equation 1, above.

The match function for the MSD classifier is given by

$$m = \sum_{i=0}^{d-1} (r_i - y_i)^2 = \bar{d} \cdot \bar{d} = |\bar{d}|^2$$

We will use the following notation.

$$p_{\bar{d}}(\bar{\alpha}) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\{-\frac{1}{2} \bar{\alpha} \Sigma^{-1} \bar{\alpha}^T\}$$

This treats two special cases:

(1) For $\tau=0$, $\Sigma=\Lambda_n$, and

(2) for $\tau>L_T$, $\Sigma \cong 2\Lambda_T+\Lambda_n$.

To begin, we know that $\bar{d}$ can be transformed by an appropriate orthogonal transformation into a vector, say $\bar{u}$, that has the following properties.

(1) The components of the vector $\bar{u}$ are independently distributed, zero-mean Gaussian random variables with different variances (in general).

(2) If λ_i^2 denotes the variance of u_i, we know that λ_i^2 is the i^{th} eigenvalue of the covariance matrix Σ.

(3) The inner product $\bar{d}\cdot\bar{d}$ is preserved under the transformation so that $\bar{u}\cdot\bar{u} = \bar{d}\cdot\bar{d}$.

Now we return to some statistical ideas. We know that if u_i is distributed as $N(0,\lambda_i^2)$ then u_i^2 is distributed as a Gamma distribution, $G(\alpha,r)$, with parameters $\alpha = 1/2\lambda_i^2$ and $r = 1/2$. We know $G(\alpha,r)$ has the form $\alpha/\Gamma(r)(\alpha x)^{r-1} e^{-\alpha x} (x>0)$, hence, $p_{u_i^2}(v_i) = (2\pi\lambda_i^2 v_i)^{-1/2} \exp\{-v_i/2\lambda_i^2\}$

The distribution on u_i has the characteristic function

$$(1 - j2\lambda_i^2\omega)^{-1/2}$$

Because the u_i are independently distributed, the characteristic function of the distribution on $\bar{u}\cdot\bar{u} = m$ is given by

$$\prod_{i=0}^{d-1} (1 - j2\lambda_i^2\omega)^{-1/2}$$

Hence $$p_\tau(m) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jm\omega} \prod_{i=0}^{d-1} (1 - j2\lambda_i^2\omega)^{-1/2} d\omega.$$

This is an exact expression for the Gauss-Markov terrain model and the discrete MSD classifier. Unfortunately, it appears there is no closed form expression for the inverse Fourier transform.

Before finding an approximation for $p_\tau(m)$, we note that in the case $\lambda_i^2 = \lambda$ for all i, $p_\tau(m)$ is $G(1/2\lambda, d/2)$. We further note that in the case $\lambda_i^2 = \lambda$ for $i = 0,1,2,\ldots I-1$, and $\lambda_i^2 = 0$ for all i such that $I \leq i \leq d-1$, $p_\tau(m)$ is $G(1/2\lambda, I/2)$.

Now we want to evaluate the λ_i. Specifically, we want to know what are the eigenvalues of the covariance matrix, Σ. We do not wish to evaluate the λ_i exactly, but we do want a good approximation to their behavior. The approximation comes from noting that Σ is a Toeplitz form (Ref 7), and hence, as $d \to \infty$, Σ assymptotically approaches a circulant matrix. The eigenvalues of the Toeplitz form and of its associated circulant matrix are distributed identically in the limit. Further, the eigenvalues of a circulant matrix can be obtained exactly as the discrete Fourier transform of the first row of the matrix (Ref 7 and 5, p 205).

A Toeplitz matrix of order 8, its associated circulant matrix and the error matrix involved in the approximation are shown in Fig 2. The approximation is very good for Toeplitz matrices whose first rows contain many trailing zeroes, or whose first rows are essentially constant. This corresponds in our model to $L_T << d$ and $L_T >> d$, respectively.

The behavior of the eigenvalues of a covariance matrix having elements

$$\exp\{-|i - j|/L\}$$

as approximated by the Toeplitz theory is shown in Fig 3. Also shown in Fig 3 is a "pulse-approximation." The eigenvalues are approximated a second time by a constant value over the first N large eigenvalues, and by zero over the remaining small eigenvalues. The constant and the number, N, are chosen to minimize the mean-square-error of the pulse approximation.

(a)

$$\begin{bmatrix} t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 & 0 & 0 \\ t_1 & t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 & 0 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 \\ 0 & t_2 & t_1 & t_0 & t_1 & t_{-2} & 0 & 0 \\ 0 & 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 \\ 0 & 0 & 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ 0 & 0 & 0 & 0 & t_2 & t_1 & t_0 & t_{-1} \\ 0 & 0 & 0 & 0 & 0 & t_2 & t_1 & t_0 \end{bmatrix}$$

(b)

$$\begin{bmatrix} t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 & t_2 & t_1 \\ t_1 & t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 & t_2 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 & 0 & 0 \\ 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 & 0 \\ 0 & 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & 0 \\ 0 & 0 & 0 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_{-2} & 0 & 0 & 0 & t_2 & t_1 & t_0 & t_{-1} \\ t_{-1} & t_{-2} & 0 & 0 & 0 & t_2 & t_1 & t_0 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & t_2 & t_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_{-1} & t_{-2} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2. The circulant approximation to a Toeplitz matrix: (a) a Toeplitz matrix of order 8, (b) the associated circulant matrix, and (c) the error matrix.

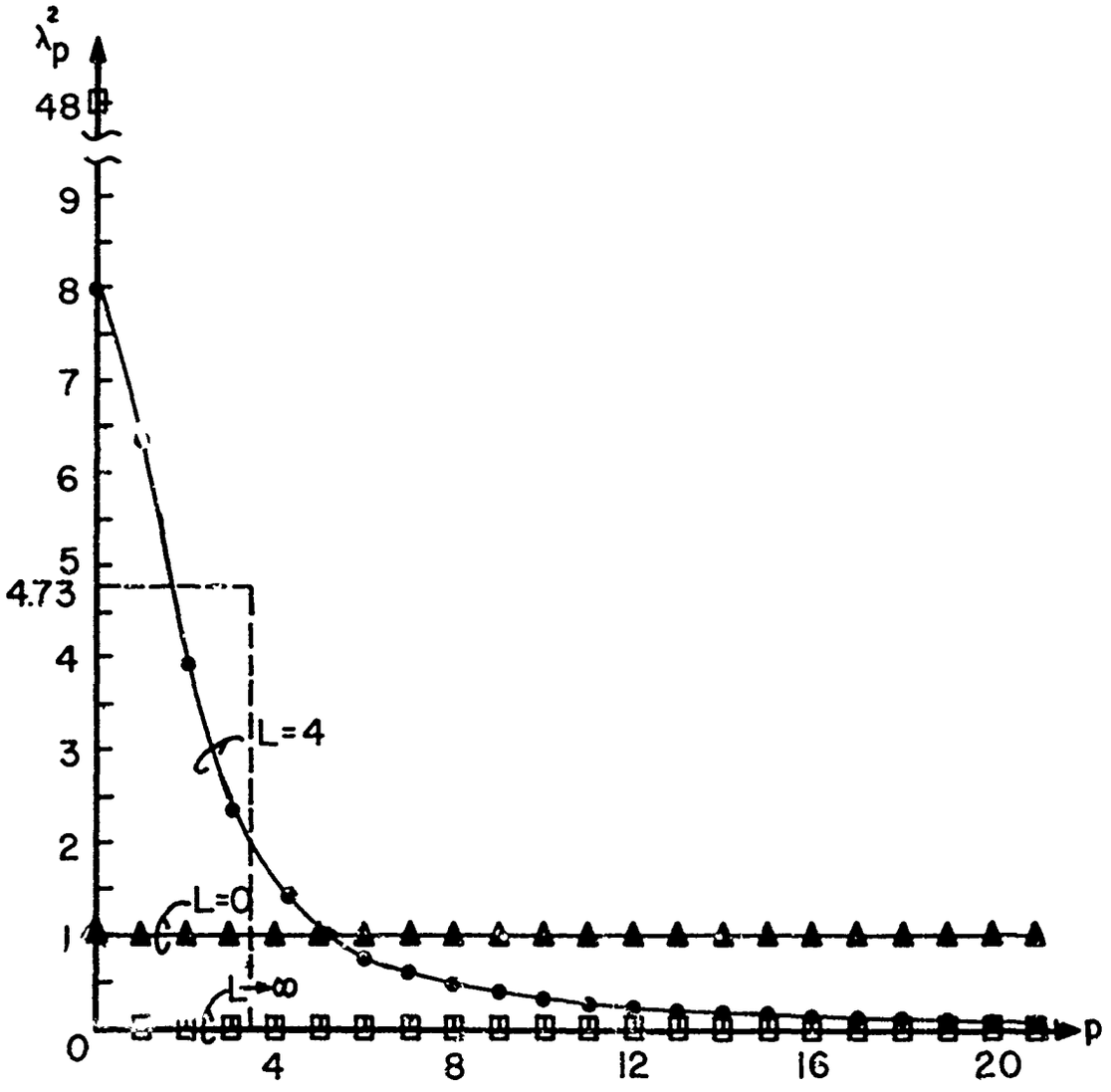


Figure 3. Asymptotic behavior of the eigenvalues of Λ with d=48, showing "pulse"-approximation for the case L=4.

Thus, we approximate $p(m_0)$ and $p_\tau(m)$ (for $\tau > L_T$) by Gamma distributions having parameters defined in terms of the eigenvalues of the covariance matrices for the terrain and the noise. The eigenvalues are approximately determined by the Toeplitz theory and the "pulse" approximation. Computer programs to implement these approximations are found in Appendix A.

We may note that this approximation yields results that agree with Schwartz (Ref 4). The eigenvalues of Σ are determined jointly by the track length, d, and by the correlation lengths, L_T and L_N. The product of d and $L_T(L_N)$ yields the β of Ref 4. The behavior of the Gamma functions derived from the approximation agree with the behavior of the approximate distributions given in Figure 2 of Ref 4.

As we shall see, the Gamma distributions can be integrated easily to obtain the false-fix probabilities for TERCOM.

THE MAD CLASSIFIER

The match function for the MSD classifier is given by

$$m = \sum_{i=0}^{d-1} |r_i - y_i| = \sum_{i=0}^{d-1} |d_i|$$

where the notation is the same as in previous sections. This match function is often referred to as the first Minkowski metric because it corresponds to K=1 in the following definition of the Minkowski metrics.

$$M_K \triangleq \sqrt[K]{\sum_{i=0}^{d-1} |r_i - y_i|^K}$$

We may note that K=2 corresponds to the Euclidean metric found in the MSD classifier discussed in the preceding section.

Because the M_1 metric is not preserved under an orthogonal transformation, it is very difficult to obtain the distribution on m from knowledge (or assumption) about the distribution on $\bar{d}$. The only case that is at all tractable is the case of independently distributed components of $\bar{d}$. This would correspond to a diagonal Σ, and that is an unwarrented assumption for TERCOM.

We can, however, gain some insight into the performance of the MAD classifier by considering a theorem in pattern recognition that states that any classification procedure is optimal (in a Bayes sense) for some distribution on the patterns. A meaningful question to ask about M_1 nearest-prototype classifiers is, therefore, for what distribution of $\bar{r}$ conditioned on $\bar{y}(\tau)$ are such classifiers optimal?

Toussaint (Ref 2) has shown that minimizing M_1 distances is equivalent to maximizing the Laplacian distribution. To make this clear, let $p(\bar{r}|\bar{y}(\tau))$ be the conditional probability of receiving (observing) $\bar{r}$ when we are truly at location $\bar{y}(\tau)$. As before, $\bar{r}$ and $\bar{y}(\tau)$ are d-dimensional vectors. If

$$p(\bar{r}|\bar{y}(\tau)) = \frac{1}{\sigma^d} \prod_{i=0}^{d-1} \exp\left[-\frac{|r_i - y_i|}{\sigma}\right] = \frac{1}{\sigma^d} \prod_{i=0}^{d-1} \exp\left\{\frac{|n_i|}{\sigma}\right\}$$

(σ a constant), then assigning $\bar{r}$ to $\bar{y}(\tau)$ on the basis of nearest M_1 distance is an optimal (Bayes) procedure. The extent to which the distribution of real-world noise approximates the Laplacian distribution is, therefore, an indicator of how close to optimal the TERCOM classifier is. While this model of the noise may not be too unrealistic, it does not help us make numerical predictions about the classifier's performance. It does, however, suggest that MAD performance and MSD performance will be approximately the same when the noise process can be modeled equally well by the Laplacian and the Gauss-Markov processes. Some additional points to consider regarding M_1 classifiers follow.

The Laplacian distribution is not "rotationally" symmetric, so the decision boundaries implemented by M_1 depend on the choice of a coordinate reference system. This may be a distinct disadvantage when modeling a physical process.

The M_1 distance has an unusual "instability" that serves to point out both the coordinate dependence referred to above, and the fact that transforms that are usually considered isometries do not preserve M_1 distances. See Figure 4.

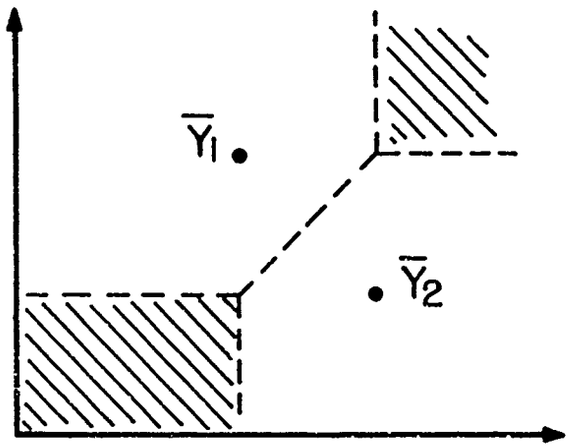


Figure 4. Decision boundaries established by an MAD classifier. The shaded areas are equidistant from $\bar{y}_1$ and $\bar{y}_2$ in the special case shown.

The MAD classifier is harder to analyze mathematically than a classifier based on Euclidean distance. This is so for two reasons: (1) An MAD classifier has piecewise linear (e.g. nonlinear) decision boundaries, whereas the Euclidean classifier has linear decision boundaries. (2) The Gaussian distribution for which the Euclidean classifier is optimal has been extensively studied in the literature due to the tractability of its functional form, whereas the Laplacian distribution does not enjoy such rich development. However, as Figure 5 shows, the difference between the two classifiers is not great in the local regions of the signal space that are "near" the signals (prototypes). Hence, an analysis of the Euclidean classifier may produce a good approximate analysis of the MAD classifier.

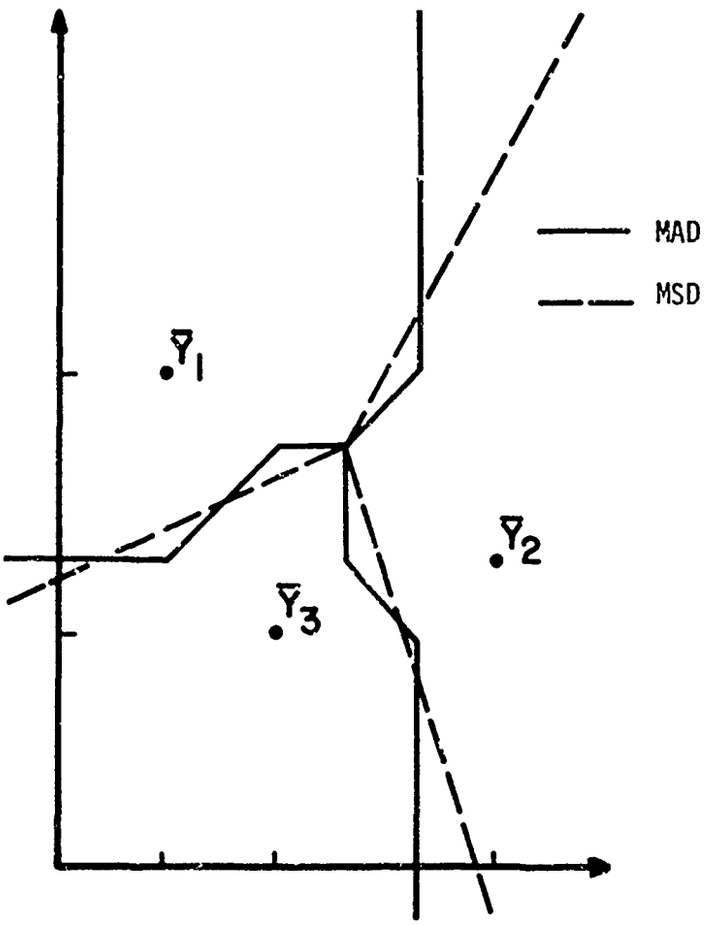


Figure 5. Comparison of decision boundaries established by an MAD and an MSD classifier.

PREDICTION OF FALSE-FIX ERROR

At this point we want to evaluate Eq 1 by using the Gamma approximation discussed earlier. We therefore assume that

$$p(m_0) = \frac{1}{2\sigma_N^2 \Gamma(I)} \left(\frac{m_0}{2\sigma_N^2}\right)^{I-1} \exp\{-m_0/2\sigma_N^2\}$$

and $$p_{\tau>L_T}(m) = \frac{1}{2\sigma^2\Gamma(K)} \left(\frac{m}{2\sigma^2}\right)^{K-1} \exp\{-m/2\sigma^2\}$$

where σ_N^2, σ^2, I, and K are the parameters involved in the approximation. (See Appendix A for a computer program to find these once the terrain and noise processes are specified.)

Thus,

$$\int_0^\infty p(m_0) \int_0^{m_0} p_{\tau>L_T}(m)\, dm\, dm_0$$

$$= \int_0^\infty \frac{1}{2\sigma_N^2\Gamma(I)} \left(\frac{m_0}{2\sigma_N^2}\right)^{I-1} \exp\{-m_0/2\sigma_N^2\}$$

$$\cdot \int_0^{m_0} \frac{1}{2\sigma^2\Gamma(K)} \left(\frac{m}{2\sigma^2}\right)^{K-1} \exp\{-m/2\sigma^2\}\, dm\, dm_0$$

By changing variables, $u = m/2\sigma^2$ and $v = m_0/2\sigma_N^2$, we rewrite this as

$$\int_0^\infty \frac{1}{\Gamma(I)} v^{I-1} e^{-v} \int_0^{(\sigma_N^2/\sigma^2)v} \frac{1}{\Gamma(K)} u^{K-1} e^{-u} \, du \, dv$$

$$= \int_0^\infty \frac{1}{\Gamma(I)} v^{I-1} e^{-v} \left[1 - e^{-(\sigma_N^2 v/\sigma^2)} \sum_{i=0}^{K-1} \frac{(\sigma_N^2 v/\sigma^2)^{K-1-i}}{(k-1-i)!} \right] dv$$

$$= 1 - \frac{1}{\Gamma(I)} \sum_{i=0}^{K-1} \frac{(\sigma_N^2/\sigma^2)^{K-1-i}}{(K-1-i)!} \int_0^\infty v^{I-1+K-1-i} e^{-(1+\sigma_N^2/\sigma^2)v} \, dv$$

$$= 1 - \frac{1}{\Gamma(I)} \sum_{i=0}^{K-1} \frac{(\sigma_N^2/\sigma^2)^{K-1-i}}{(K-1-i)!} \cdot \frac{(I+K-i-2)!}{(1+\sigma_N^2/\sigma^2)^{I+K-i-1}}$$

$$= 1 - \left(\frac{R}{R+1}\right)^I \sum_{k=0}^{K-1} \binom{I-1+k}{k} (R+1)^k$$

with $R=\sigma^2/\sigma_N^2$ and $\binom{I-1+k}{k}$ a binomial coefficient. Recall that I and K are half the number of nonzero eigenvalues of Λ_N and of $2\Lambda_T+\Lambda_N$, respectively, as derived from the approximation.

Appendix A also gives a program to compute this finite sum. A recursion formula is used to avoid an overflow problem that might otherwise arise from the factorials involved.

Once the sum is evaluated, Eq 1 is easy to compute. If the sum evaluates to a quantity less than one-half, the Prob $\{m_0 <$ all m$\}$ is set to zero.

We return to a discussion of the predictions from this model after we discuss the simulation. At that time we can compare the results of the theory and the simulation.

TERCOM SIMULATION

GENERAL CONSIDERATIONS

For a theoretical analysis of TERCOM performance to be useful, one must have confidence that most of the results predicted by this analysis can be experimentally verified. While the ultimate test is an actual flight test over real terrain, a computer simulation of a Gaussian terrain model has proven very enlightening. Samples of Gaussian terrain were generated by a digital computer program. TERCOM runs across these terrain samples were then simulated for both MSD and MAD classifiers using an uncorrelated Gaussian process to simulate system noise from all causes. A tape of digitized real terrain amplitudes was obtained which contained a wide variety of terrain types from flat to mountainous. TERCOM runs were simulated across these samples, again using the uncorrelated Gaussian process as system noise. We found that TERCOM results from both Gaussian and real terrain were almost equivalent when a certain terrain parameter ratio, σ_Z/σ_T, was the same for both types of terrain. Certain basic conclusions about TERCOM performance can be drawn from these simulations.

OUTLINE OF COMPUTER PROGRAMS

A reference array size of 64 x 64 cells was chosen for the simulation since this would represent a realistic size for an on-board aircraft computer system. The distance between sample points (cell size) was left unspecified for the Gaussian data, but the cell size used for the real terrain samples was 400 ft.

The starting point for generating the Gaussian data was a radially symmetric autocorrelation function of the form $\Phi(x,y) = \exp\{-\sqrt{x^2+y^2}/L\}$, where L is the correlation length. Small values of L indicate rough terrain with low correlation between points, while large values of L indicate smoother terrain with a high correlation between adjacent points. A two-dimensional Fourier transform of the autocorrelation function was computed using a fast Fourier transform algorithm. This procedure generated

the power spectrum, $P^2(\omega_x,\omega_y)$, of the terrain. Generation of the actual terrain, however, requires an inverse transform of the complex spectrum. There are an infinity of complex spectra for a given power spectrum, so any complex spectrum, $\Phi(\omega_x,\omega_y) = R_e(\omega_x,\omega_y) + jI_m(\omega_x,\omega_y)$, with the power spectrum $P^2(\omega_x,\omega_y) = \Phi^*\Phi$ would do, provided it has conjugate symmetry. If $\Phi(\omega_x,\omega_y) = R_e(\omega_x,\omega_y) + jI_m(\omega_x,\omega_y)$, conjugate symmetry implies that $R_e(\omega_x,\omega_y) = R_e(-\omega_x,-\omega_y)$ and $I_m(\omega_x,\omega_y) = -I_m(-\omega_x,-\omega_y)$. This is necessary and sufficient to ensure that the Fourier transform of $\Phi(\omega_x,\omega_y)$ will give a real function $T(x,y)$ representing the terrain.

Each real point of the spectrum, $R_e(\omega_x,\omega_y)$, is picked at random from a Gaussian distribution with mean zero and variance equal to $P(\omega_x,\omega_y)/3$. If $|R_e(\omega_x,\omega_y)| > P(\omega_x,\omega_y)$ a new number is picked until a value $|R_e(\omega_x,\omega_y)| \leq P(\omega_x,\omega_y)$ is found. It follows that the imaginary part of the spectrum, $I_m(\omega_x,\omega_y)$, is given by $\sqrt{P^2(\omega_x,\omega_y) - R_e^2(\omega_x,\omega_y)}$. The sign of $I_m(\omega_x,\omega_y)$ is chosen at random to be + or - with probability 1/2 for each case. When quadrants 1 and 2 of the 64 x 64 spectrum array have been computed in this way, quadrants 3 and 4 are generated from 1 and 2 using the conjugate symmetry conditions. The Fourier transform of this spectrum produces the Gaussian terrain used in the simulations of TERCOM.

Figure 6 shows a number of amplitude density plots from this terrain superimposed on Gaussian density functions having the same mean and variance. The agreement is quite good. Similarly, good agreement is shown in Figure 7, where the distribution function for the terrain data is plotted on a special normal probability versus amplitude graph paper. Gaussian distribution functions plot as straight lines on this type of graph. The steeper the slope the smaller the variance. The data is fitted quite well by a straight line. Figure 8 shows two computer plots of terrain generated by the transform method just described. In Figure 8a the correlation length, L, is 4 cells while in Figure 8b it is 8 cells. Increased correlation length has produced flatter, smoother terrain. Figures 6, 7 and 8 illustrate that the method produces terrain-like surfaces with Gaussian amplitude densities, and that terrain roughness

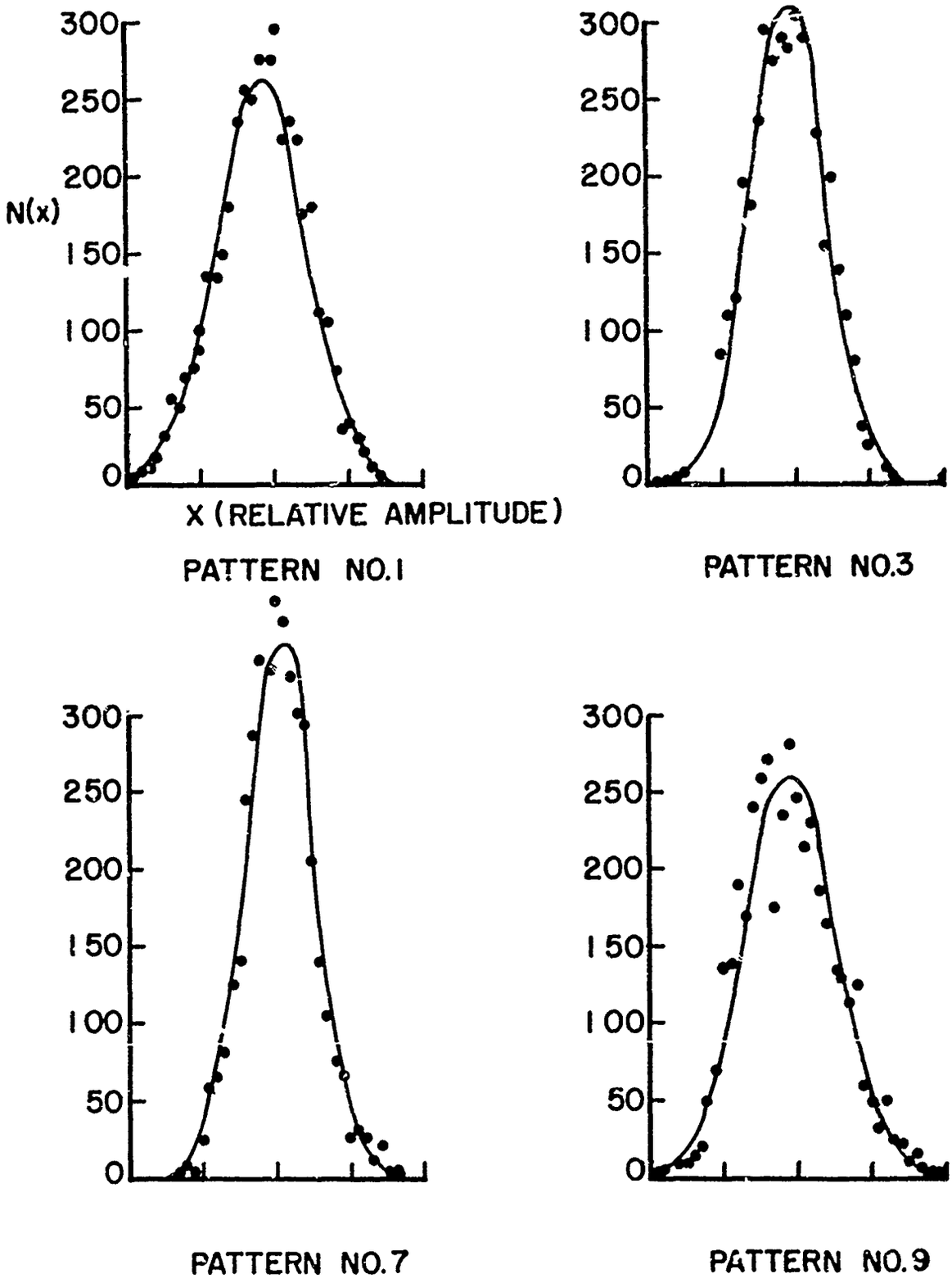


Figure 6. Gaussian terrain amplitude histograms compared with normal density curves having the same mean and variance.

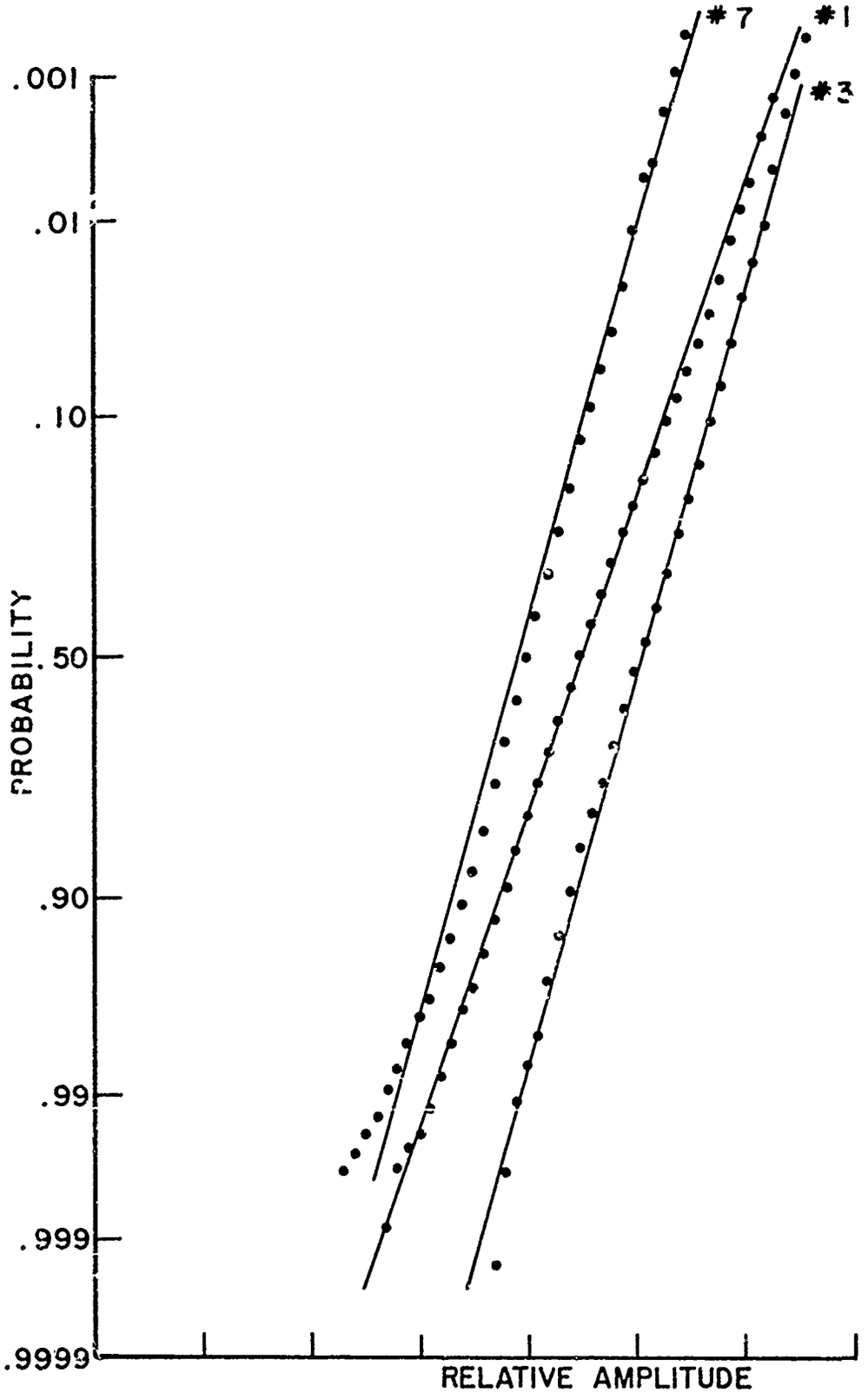
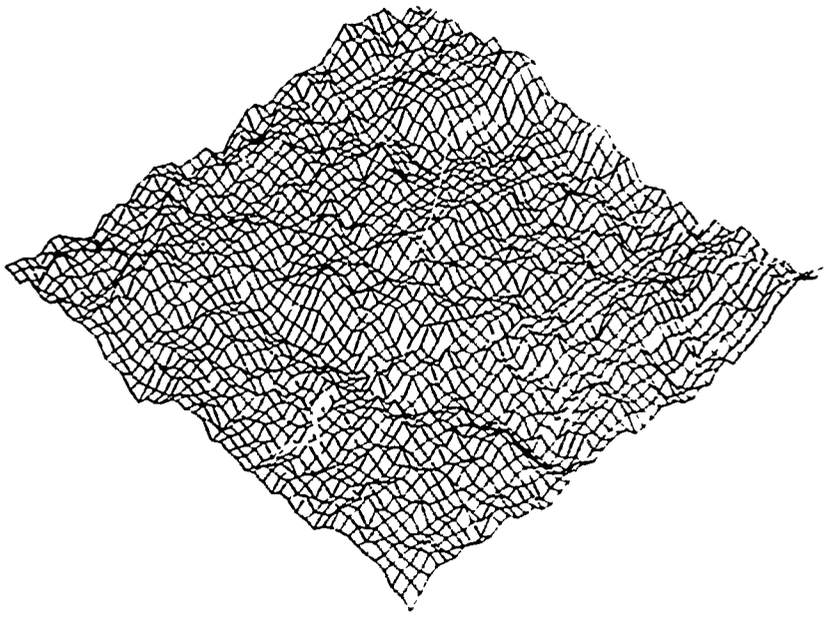
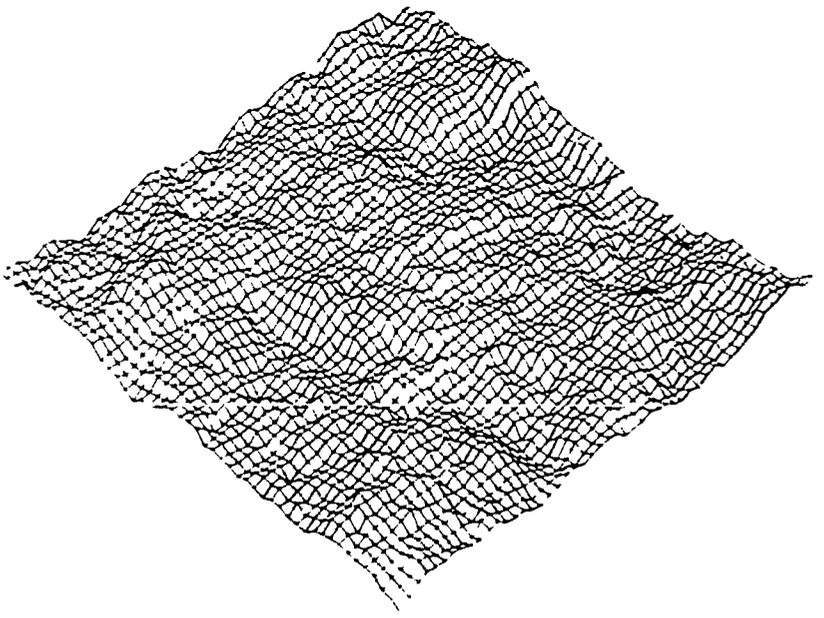


Figure 7. Probability plot of 64 x 64 Gaussian data arrays.



(a)



(b)

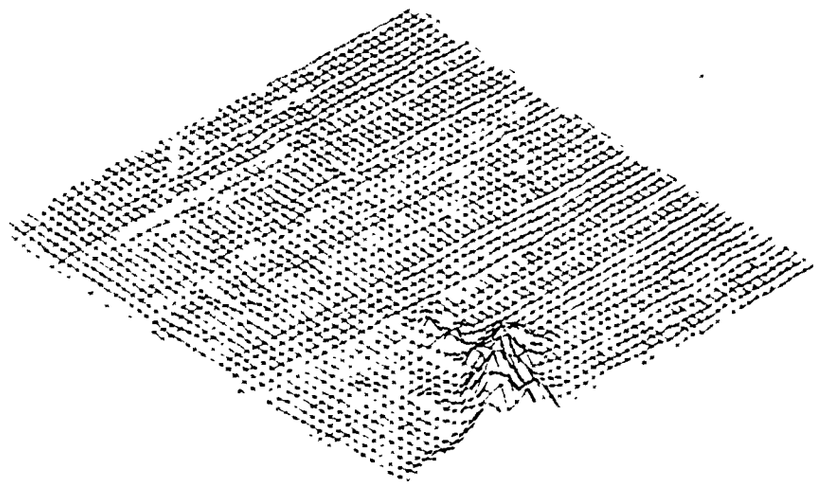
Figure 8. Samples of computer-generated Gaussian terrain. (a) L = 4; (b) L = 8.

is controlled to a certain extent by L, the correlation length. The program (TERC 1) listed in Appendix A is set up to generate and store on a permanent file 10 such (different) reference arrays of dimension 64 x 64.

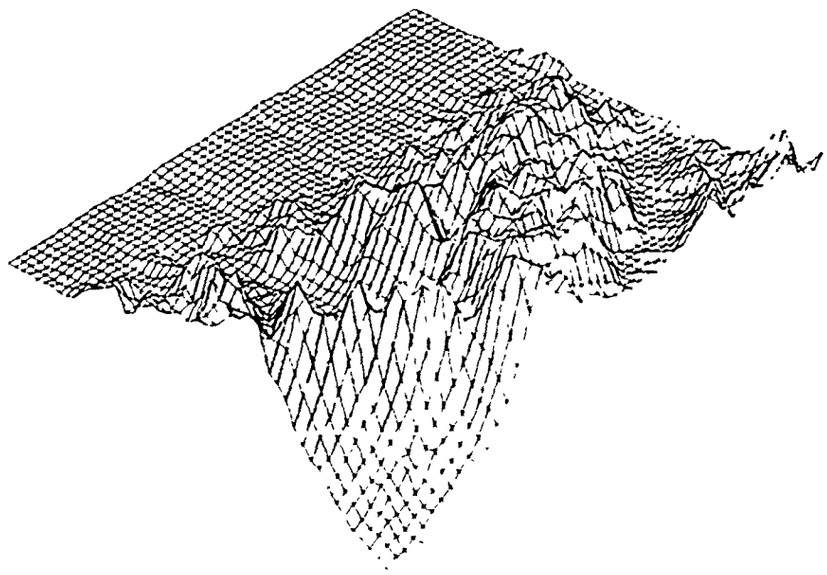
The real terrain tape contained amplitudes from a large continuous area sampled into an array of 1073 x 282 cells with a cell width of 200 ft. We chose to use 400-ft cells , sampling every other point, and divided the area into eight 64 x 64 arrays representing a variety of terrain types. These were stored on a permanent file in the same format as the Gaussian terrain, so both could be used as input to the TERCOM simulation program. Some of the real terrain samples are shown in Figures 9 and 10. Amplitude distribution and density functions were computed from the entire 1073 x 282 array. The distribution function is shown in Figure 11 plotted on a normal probability graph. The primary deviation from a straight line is below .05 probability. Thus 5% of the amplitudes deviate considerably from Gaussian behavior. The Gaussian density function and an amplitude histogram of the real terrain are shown in Figure 12. The real terrain histogram is narrower in the peak and wider in the skirts than a Gaussian density. We show, however, that the Gaussian approximation is accurate enough for TERCOM performance estimation purposes.

SIMULATION OF TERCOM SYSTEM NOISE

Simulation of the TERCOM system required testing with many levels of additive Gaussian noise. The Fourier transform method used to generate the terrain could also have been used to generate this system noise if the computer memory requirements for transforming a 64 x 64 array were not so large. Computer turnaround time on our CDC 6600 system is largely determined by memory allocation, so a more efficient method of generating correlated Gaussian noise was sought. Moshman (Ref 3) has demonstrated a method for generating a one-dimensional string of Gaussian random numbers with any desired correlation coefficient. This method was expanded to two dimensions for generating a 64 x 64 array of correlated, normally distributed amplitudes. The method is illustrated in Appendix B. Noise characteristics may be specified by correlation coefficient or correlation length. Both terrain and noise are generated from Gaussian random processes of mean zero, but then have independent variances and correlation lengths. Most of the situations studied used noise with zero correlation, but the option exists in the TERCOM simulation program to use correlated noise.

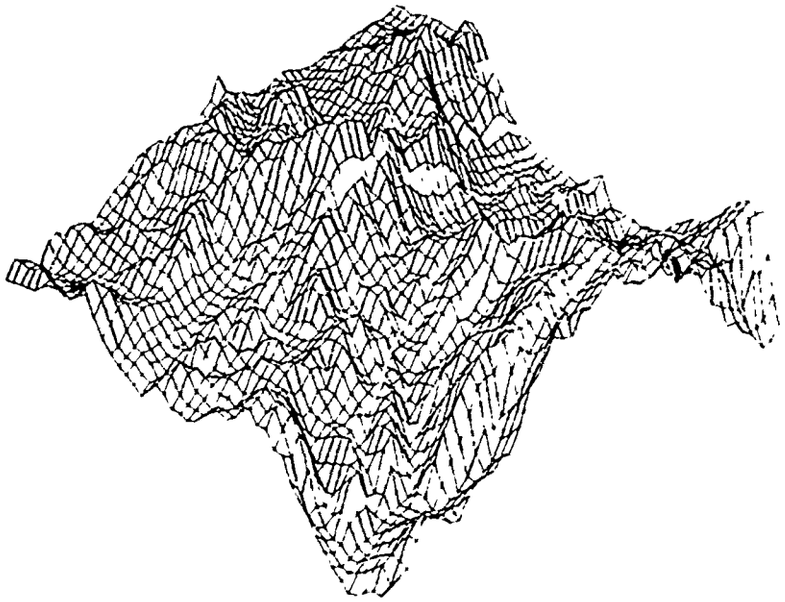


(a)

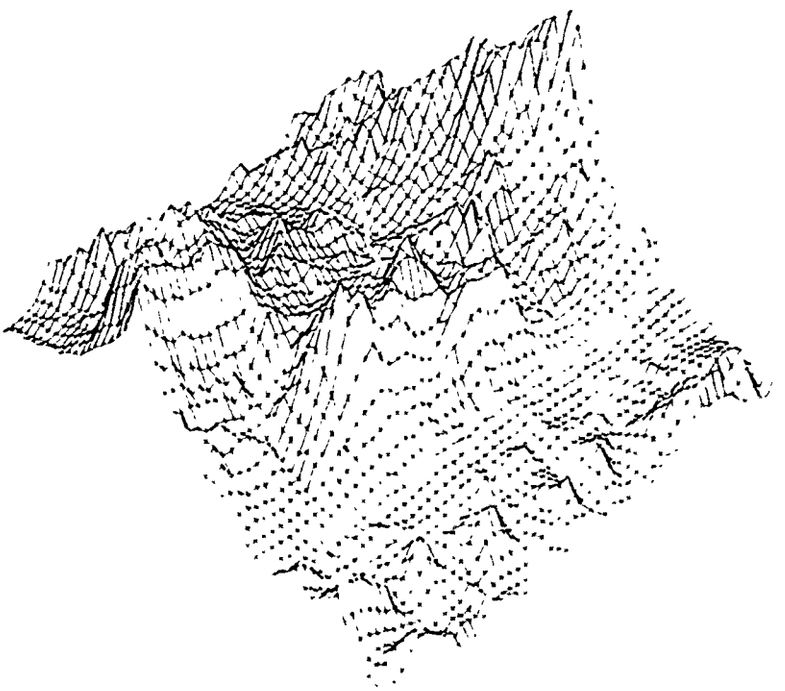


(b)

Figure 9. Samples of real terrain. (a) Area #2; (b) Area #4.



(a)



(b)

Figure 10. Samples of real terrain. (a) Area #8; (b) Area #5.

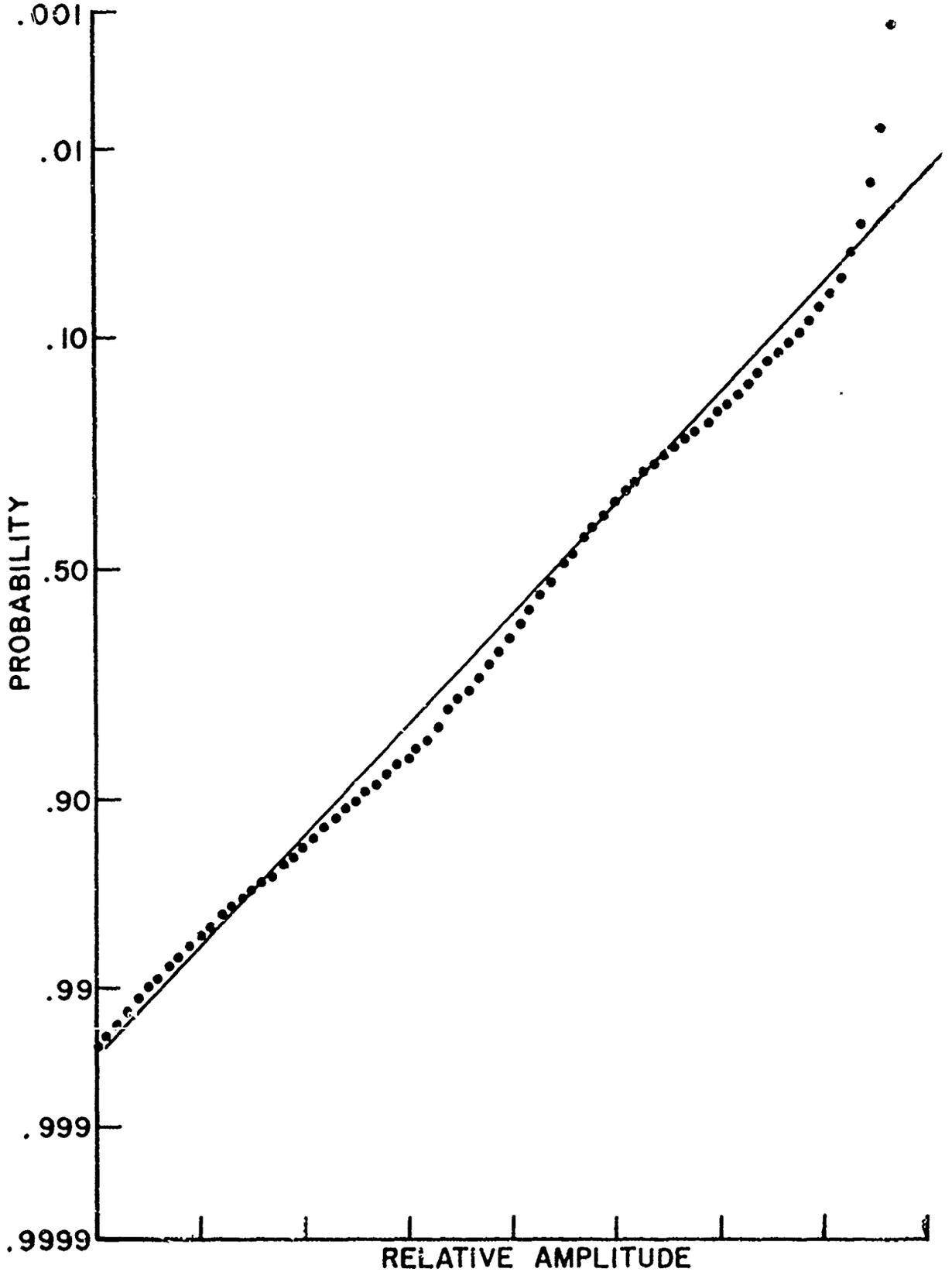


Figure 11. Probability plot of real terrain.

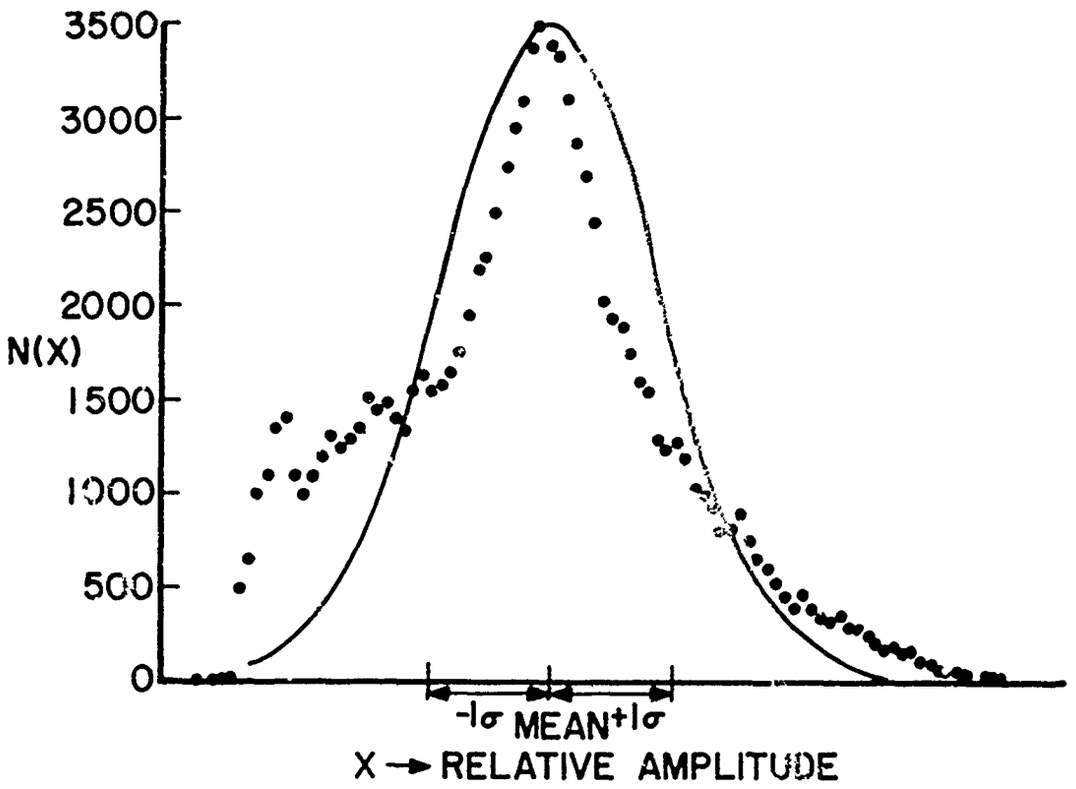


Figure 12. Real terrain amplitude histogram compared with a normal density curve having the same mean and variance.

SIMULATION PERFORMANCE ON GAUSSIAN TERRAIN

Performance of the TERCOM system on the Gaussian terrain will be discussed in some detail. It will be seen later that TERCOM performance on real terrain seems to be somewhat different, but Gaussian terrain performance can be transformed into a close approximation to real terrain performance. Therefore, the conclusions made in this report with regard to the Gaussian terrain will be seen to hold also for real terrain.

We have assumed that the form of additive noise is Gaussian in the absence of any other evidence. Previous reports by LTV E-Systems have indicated that a large source of noise is the error made in transferring measurements from maps or aerial photographs to the digitized on-board memory of the TERCOM system. Such errors are likely to be Gaussian in nature. We have lumped this together with any sensor noise into one Gaussian function in our model. Total system noise is varied by changing the variance (σ^2_N) of this noise function.

The TERCOM system in an aircraft scans a strip of terrain over which it is flying, recording and digitizing terrain altitudes at regular intervals until some predetermined number (d) have been recorded. The on-board memory contains a digitized representation of the surrounding terrain area over which the aircraft is expected to be flying when sampling starts. This digitized terrain is in the form of an M by N matrix in which both M and N are usually larger than d. We will assume that the aircraft is flying over the represented terrain in a columnwise direction. The distance will then be computed between the d-dimensional scanned vector and all the column subvectors of size d contained in the array. These (M-d+1)N distances are computed by either a mean square distance algorithm (MSD) or a mean absolute difference algorithm (MAD). In each case the array vector with minimum distance from the scanned vector is picked as a match and the navigation system uses the coordinates of this match to fix its position.

Our computer simulation used a 64 x 64 array of amplitudes to represent the terrain. This same array, with Gaussian noise added, then represented the on-board memory. Scanning track length for most cases was

chosen to be 48 cells. We assumed that scanning would start inside the specified area and be completed within this area. No study was made of the situation where scanned tracks started or stopped outside the area of interest. Thus, in our case M and N were 64; track starting coordinates were 1 to 64 in the N direction and 1 to 17 in the M direction ($M - d + 1 = 64 - 48 + 1 = 17$). For each terrain sample, 10 such starting points were chosen from a uniform distribution over the 64 x 17 area. A single computer run produced 10 different terrains with 10 scanned tracks per terrain, giving 100 trials to the TERCOM system at a given signal to noise ratio. The signal to noise ratio was specified as σ_T/σ_N where σ_T is the standard deviation of the terrain and σ_N is the standard deviation of the noise.

TERCOM performance using both MSD and MAD classifiers is shown, for Gaussian terrain, in Figure 13. Here terrain correlation length is four cells and system noise is uncorrelated. The curves represent the mean of 400 TERCOM trials at each signal to noise ratio. One striking feature of these curves is that system performance never reaches 100% correct identification. The average miss distance also remains essentially constant. This failure to reach 100% correct at high signal to noise ratios follows from the method of "mean removal" used by TERCOM to normalize the terrain data.

Mean removal in the present TERCOM system is carried out as follows.

(1) The mean of the stored 64 x 64 array is computed and subtracted from each element of the array to assure that the stored array has zero mean.

(2) The mean of the scanned strip is computed and subtracted from each component of the strip vector to assure that the scanned strip has mean zero.

(3) The zero mean strip vector is compared with equal length strips in the zero mean stored array using the MSD or MAD algorithm.

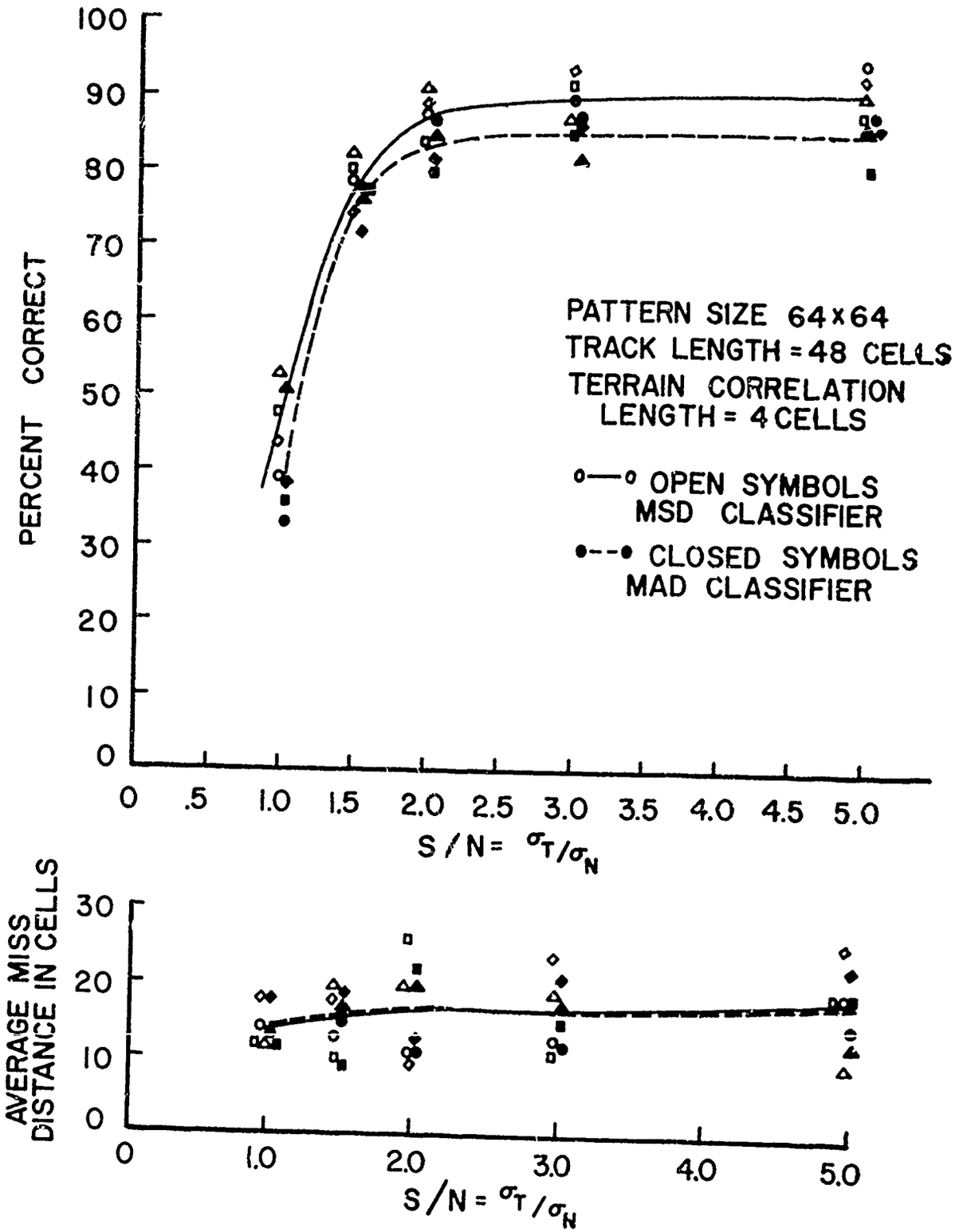


Figure 13. TERCOM performance on Gaussian terrain with uncorrelated Gaussian noise.

The problem with this method is in step 1 above. The system computes the distance between a d-dimensional zero mean scanned strip vector and a d-dimensional vector picked from an array with mean zero but with dimensionality many times larger than d. A small sample from an array of points with mean zero will in general have nonzero mean. Thus, the distance between the two vectors will almost never be zero, even in the zero noise case. This problem can be remedied by simply removing the mean of each strip vector chosen from the array before the distance from the zero mean scanned strip is computed. This alteration gives the results shown in Figure 14, where the terrain used is the same as that which produced Figure 13. The percentage of correct identifications rises quickly to 100%, and the average miss distance falls quickly to zero. We see that the simulation has pointed out a procedural error not too obvious from purely theoretical studies.

A number of computer runs were made to check out various effects such as correlated noise and variations in track length. These runs were made with the original TERCOM system which could not reach 100% correct performance. The curves have generally the same shape in all cases. Correlated noise with correlation length the same as the terrain causes a noticeable drop in performance, but the average miss distance is reduced somewhat. These results are shown in Figure 15. The percentage of correct identifications was maximum for uncorrelated system noise. TERCOM performance with this type of noise, using both MSD and MAD classifiers, is shown in Figure 16a. Here the percentage of correct identifications is shown as a function of track length and terrain correlation length. In all cases MSD is slightly superior to MAD (this is to be expected - we forced the process to be Gaussian). Average miss distance as a function of track length and terrain correlation length is shown in Figure 16b.

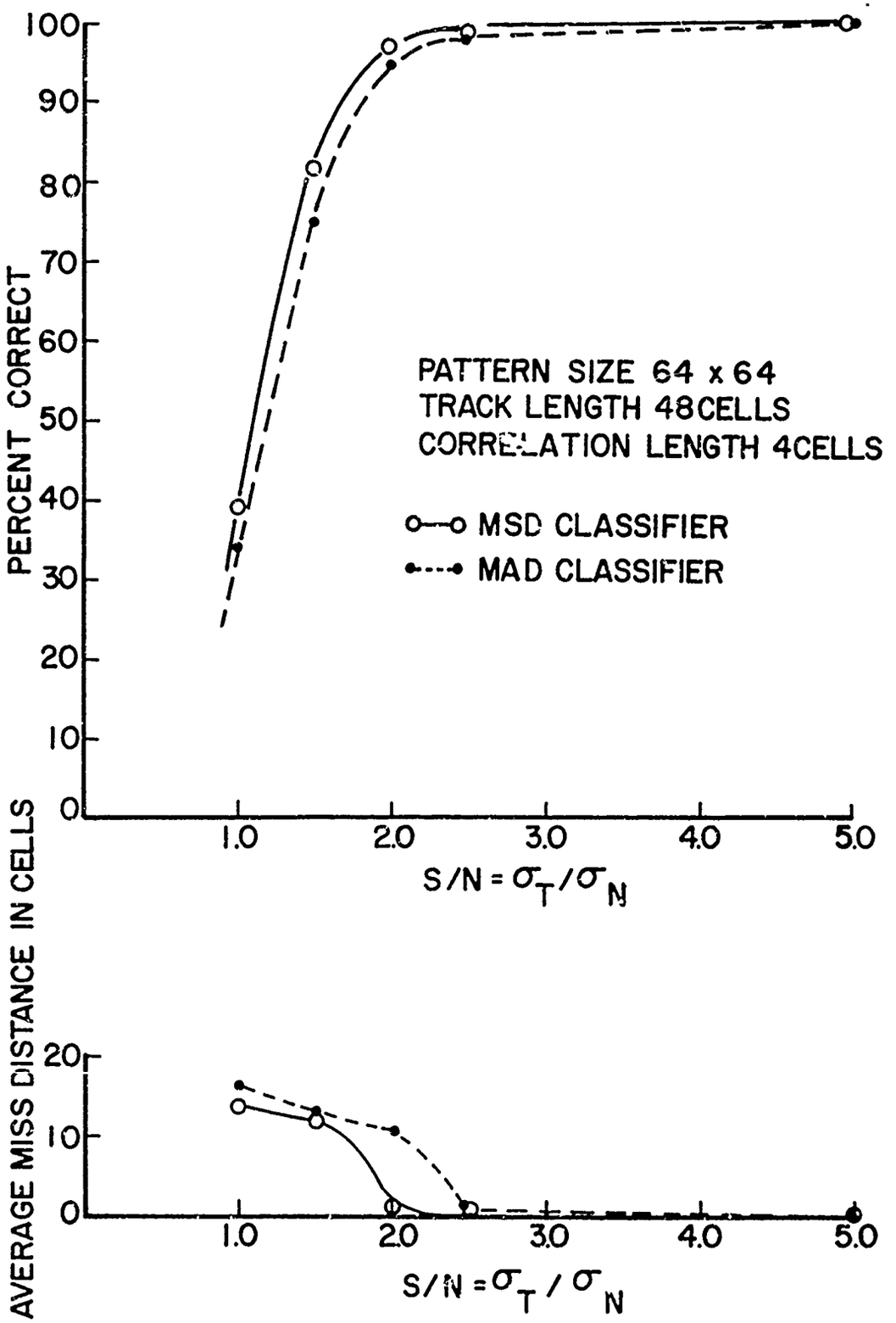


Figure 14. TERCOM performance with improved mean removal and uncorrelated Gaussian noise.

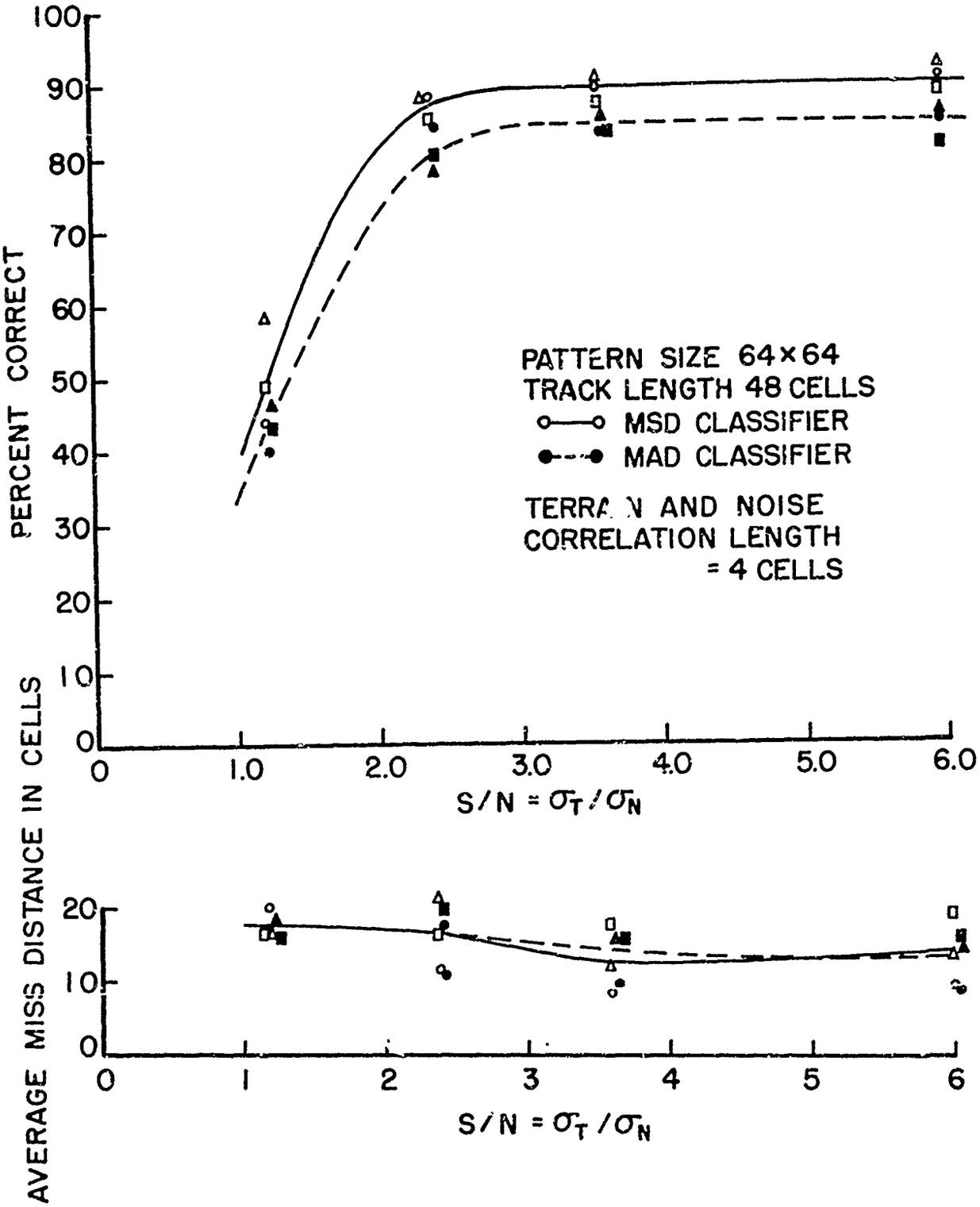
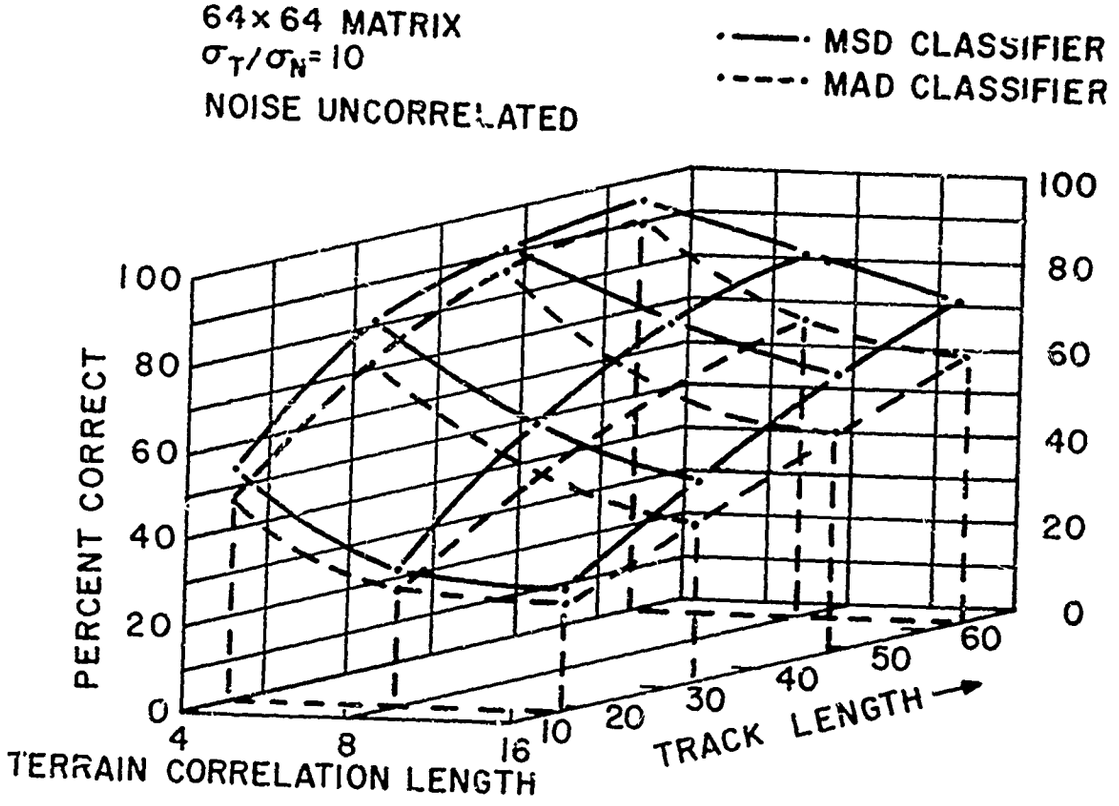
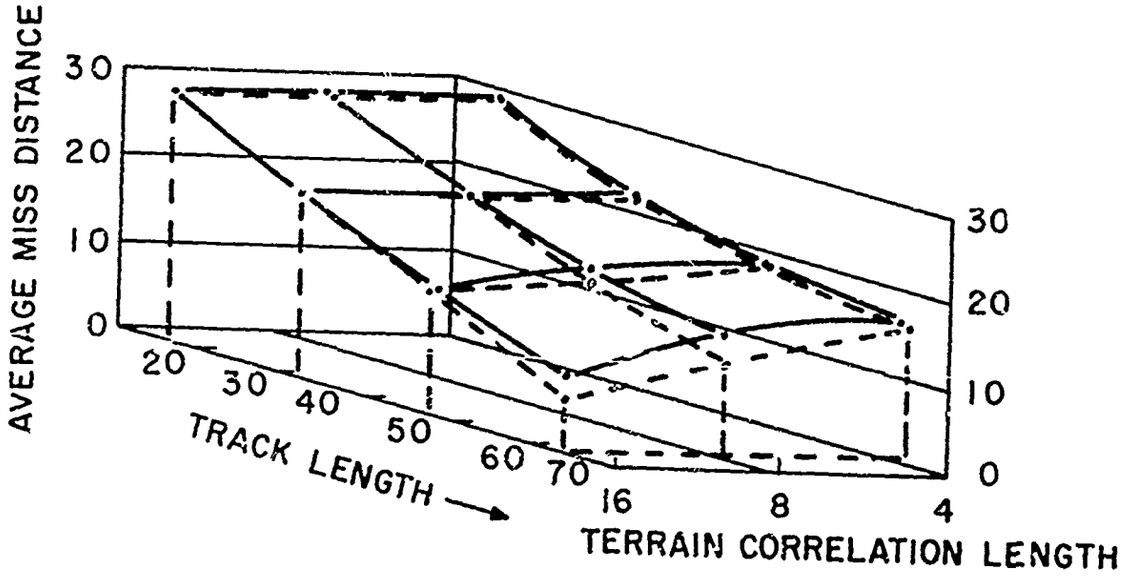


Figure 15. TERCOM performance with correlated system noise.



(a)



(b)

Figure 16. TERCOM performance as a function of track length and terrain correlation length. (a) percent correct; (b) average miss distance.

One more study involving terrain correlation length was done using the improved TERCOM system. Performance curves were plotted for increasing terrain correlation lengths, and some of the results are shown in Figure 17. We see that the curves approach an asymptote as the correlation length is increased. This asymptotic curve is important in interpreting the results of the next section, where TERCOM performance on real terrain is considered.

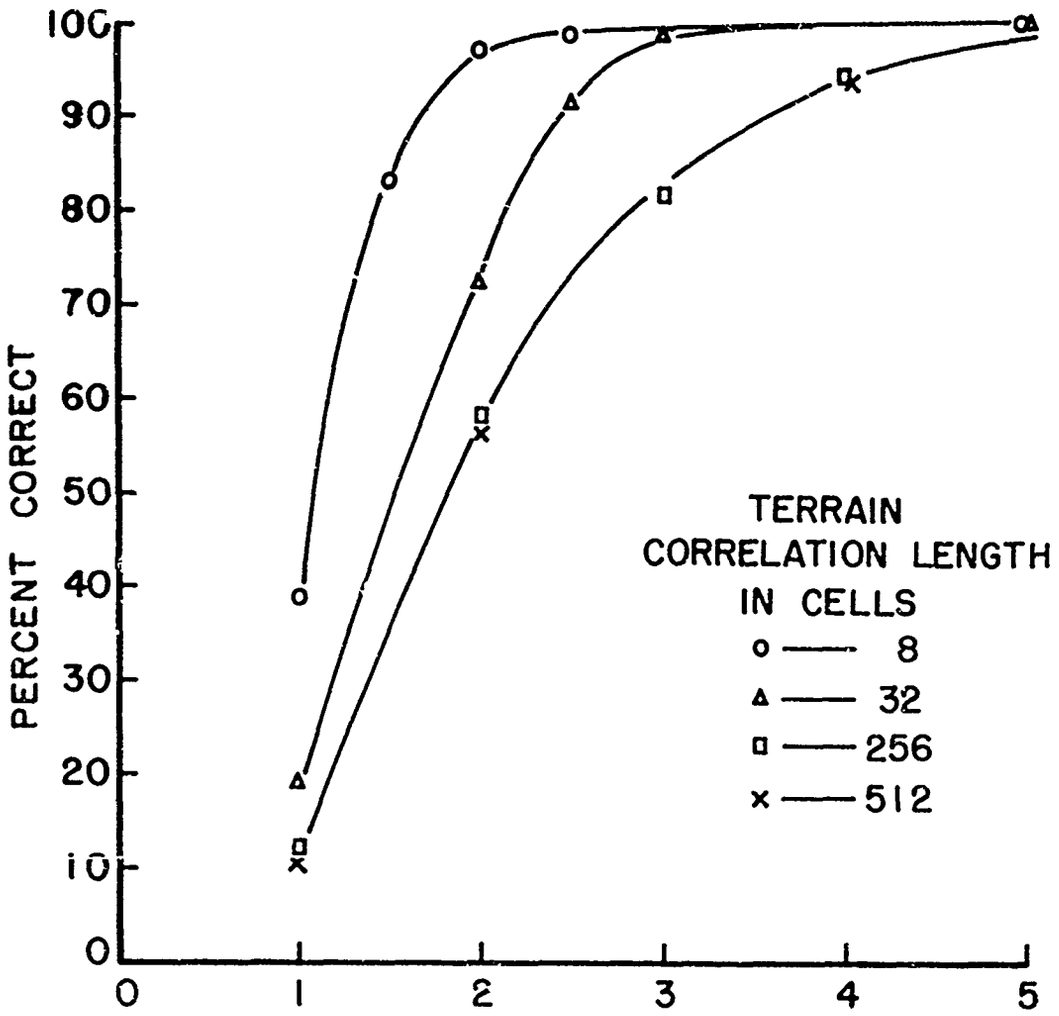


Figure 17. TERCOM performance on Gaussian terrain, showing asymptotic behavior with increasing correlation length. Only the MSD classifier with improved mean removal is shown.

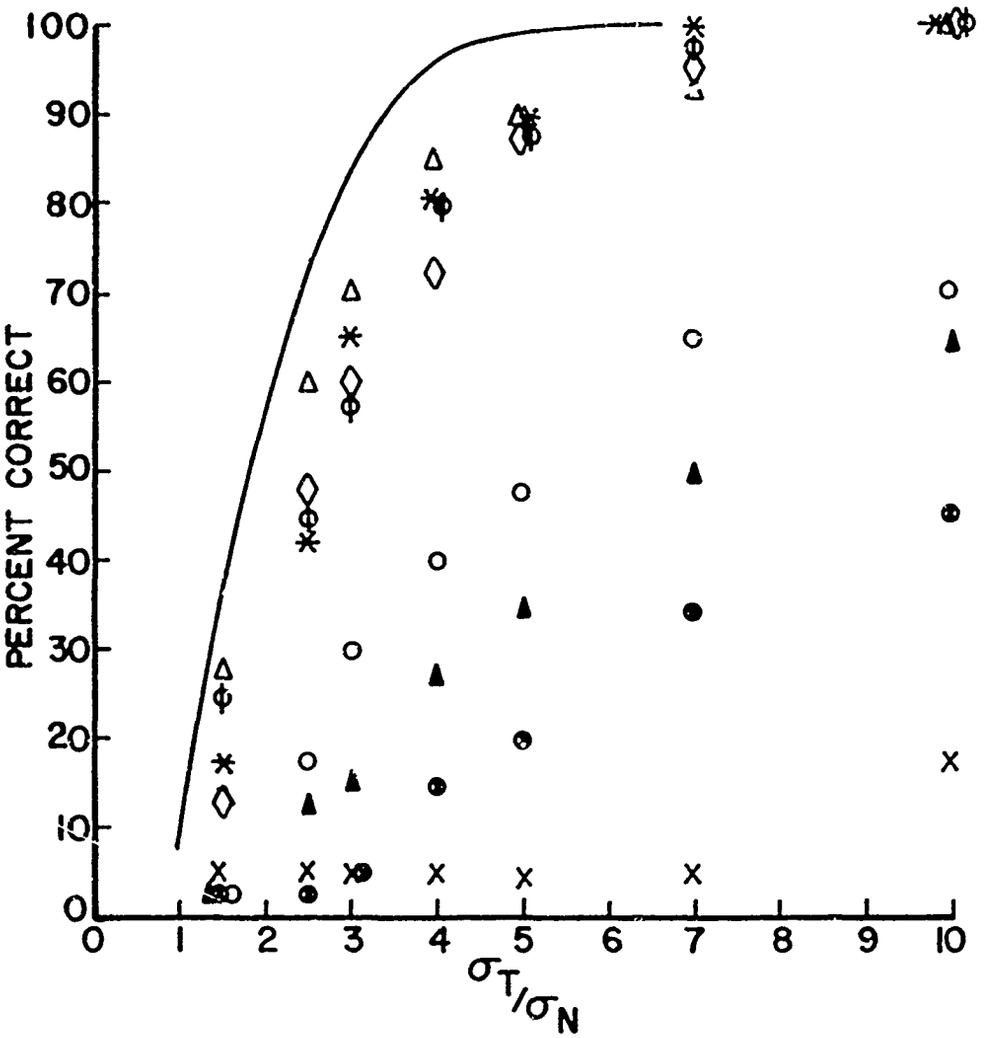
SIMULATION PERFORMANCE ON REAL TERRAIN

Each of the eight real terrain patterns was used for 40 TERCOM runs at each of seven signal to noise ratios. The results of these runs are shown as the data points of Figure 18. Each point represents the percentage of correct matches in 40 TERCOM runs.

Note that all of the points lie below the solid curve in the figure. This curve represents the performance asymptote of Figure 17 approached by TERCOM trials on Gaussian terrain as correlation length is increased. Thus we see a wide variety of curves for TERCOM applied to real terrain, all lying below the lowest possible performance curve produced on Gaussian terrain. Examination of computer plots of real terrain, such as those shown in Figures 9 and 10, revealed a correlation between initial slope of the performance curves and the amount of flat area in the terrain over which this performance was computed. Pattern 1 was completely flat. Patterns 2, 3, and 4 showed increasing amounts of mountainous area. Patterns 5, 6, 7 and 8, which cluster together, were almost all completely mountainous. These last four are also the curves nearest the asymptotic performance curve for the Gaussian terrain.

A quick analysis of the TERCOM algorithm will make it obvious that TERCOM cannot operate successfully on a nearly flat or planar area, and we should expect more mistakes on terrain containing flat areas. However, the data for σ_T on the right of Figure 18 show that there is no apparent correlation between σ_T and the amount of flat or planar area present in the pattern. An area may be planar but have non-zero slope and consequently a fair sized σ_T. Thus σ_T and the ratio σ_T/σ_N are not the correct parameters to characterize performance over real terrain. Some measure of terrain roughness seems to be needed. Ideally, it seems that this should be some sort of second derivative or second difference. A simpler measure, which seems to work well for real terrain is

$$\sigma_z^2 = E\{(x_i - x_{i+1})^2\}$$



PATTERN	SYMBOL	σ_T
1	×	4.4
2	⊛	9.5
3	▲	14.6
4	○	16.5
5	△	11.7
6	◇	12.3
7	✳	13.6
8	ϕ	14.8

— ASYMPTOTIC BEHAVIOR OF TERCOM ON GAUSSIAN TERRAIN AT A CORRELATION LENGTH OF 1024 CELLS

Figure 18. Correct match scores for MSD classifier over real terrain as a function of σ_T/σ_N.

This parameter gives a measure of the average difference between adjacent sample points in the direction of the expected flight path. The ratio σ_z/σ_T was found to be especially useful in characterizing the terrain. A small value of σ_z/σ_T indicates smooth terrain with small variation between sample points, but possible large slow fluctuations in amplitude over the whole area. Large σ_z/σ_T indicates variation between adjacent sample points is large compared to the overall amplitude variation in the area.

The ratio σ_z/σ_T was computed for the Gaussian terrain. Because TERCOM performance on Gaussian terrain approached the asymptote of Figure 18, the σ_z/σ_T ratio for Gaussian terrain asymptotically approached 0.3. This was higher than the σ_z/σ_T values for real terrain, which ranged from .06 for very flat areas to .22 for completely mountainous areas. The real terrain data of Figure 18 was then replotted in terms of σ_z/σ_N, where σ_N took on the values of $\sigma_T/1.5$, $\sigma_T/2.5$, $\sigma_T/3$, $\sigma_T/4$, $\sigma_T/7$ and $\sigma_T/10$. This replotted data is shown in Figure 19; it exhibits a very nice clustering of the data points. Three distinct groupings of curves are evident, and there is correlation between initial slope and σ_z/σ_T as can be seen on the right of Figure 19. Further, all curves seem to originate near the same value of σ_z/σ_N.

Flat areas were then introduced into the Gaussian terrain in the hope that this would decrease their σ_z/σ_T below the asymptotic value of 0.3. The flat areas were introduced by simply setting entire columns of the 64 x 64 matrix equal to zero. This had the desired effect, and three modified Gaussian arrays were generated. The number of zeroed columns and corresponding σ_z/σ_T values were: 2 columns zeroed, σ_z/σ_T = .22; 16 columns zeroed, σ_z/σ_T = .10; 32 columns zeroed, σ_z/σ_T = .06. This modification was done with Gaussian terrain of correlation length 1024 cells.

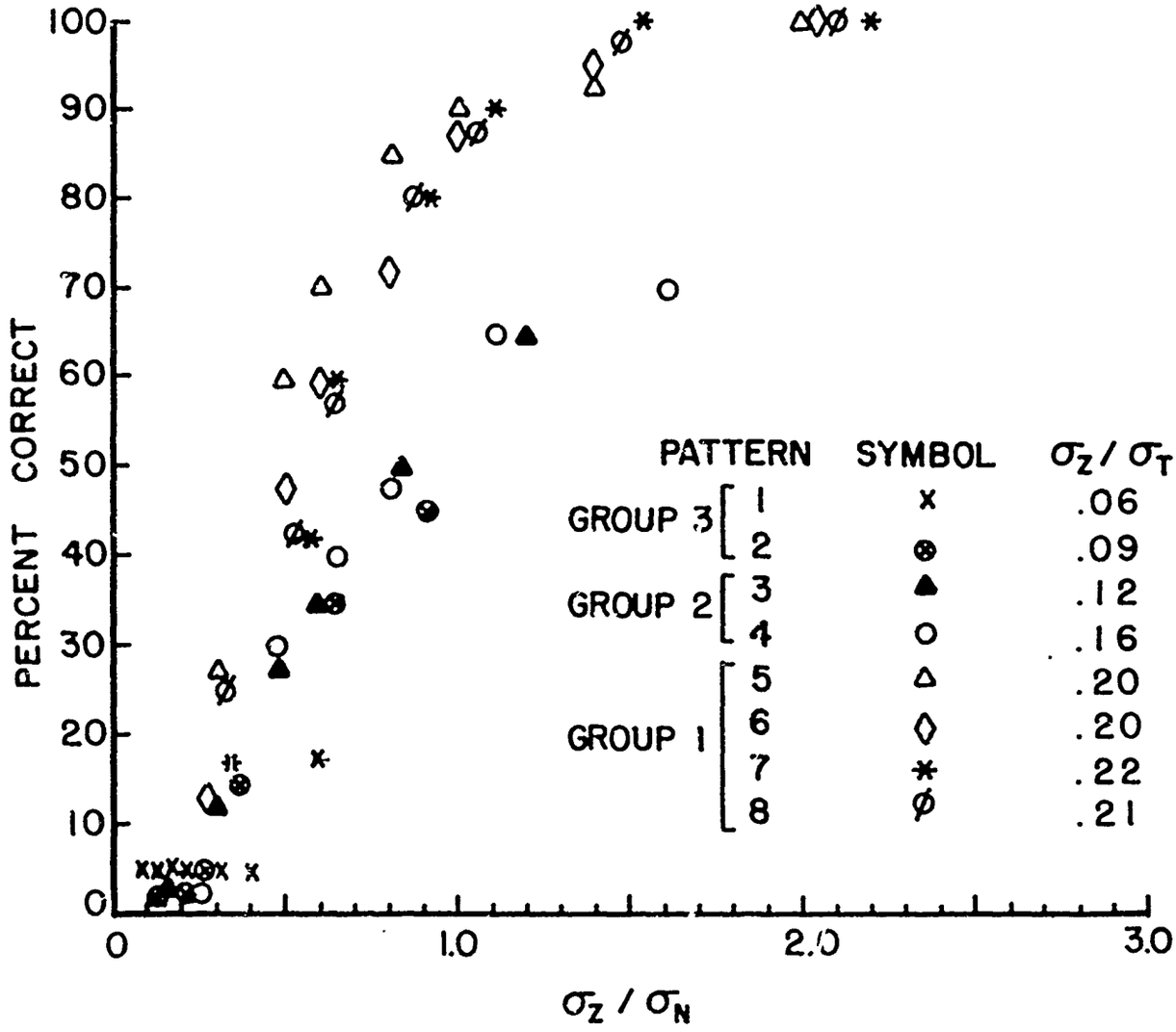


Figure 19. Correct match scores for MSD classifier over real terrain as a function of σ_z/σ_N. The data show the emergence of three groups.

The percentage of correct scores of a TERCOM classifier using the MSD algorithm on partially flattened Gaussian terrain are shown in Figures 20 and 21, along with the appropriate real terrain data points. Note that for all but one case with very low σ_z/σ_T, the Gaussian model underestimates the real terrain scores by an average of about 10%. Thus, this simply modified Gaussian terrain model gives a somewhat pessimistic prediction of real terrain TERCOM performance.

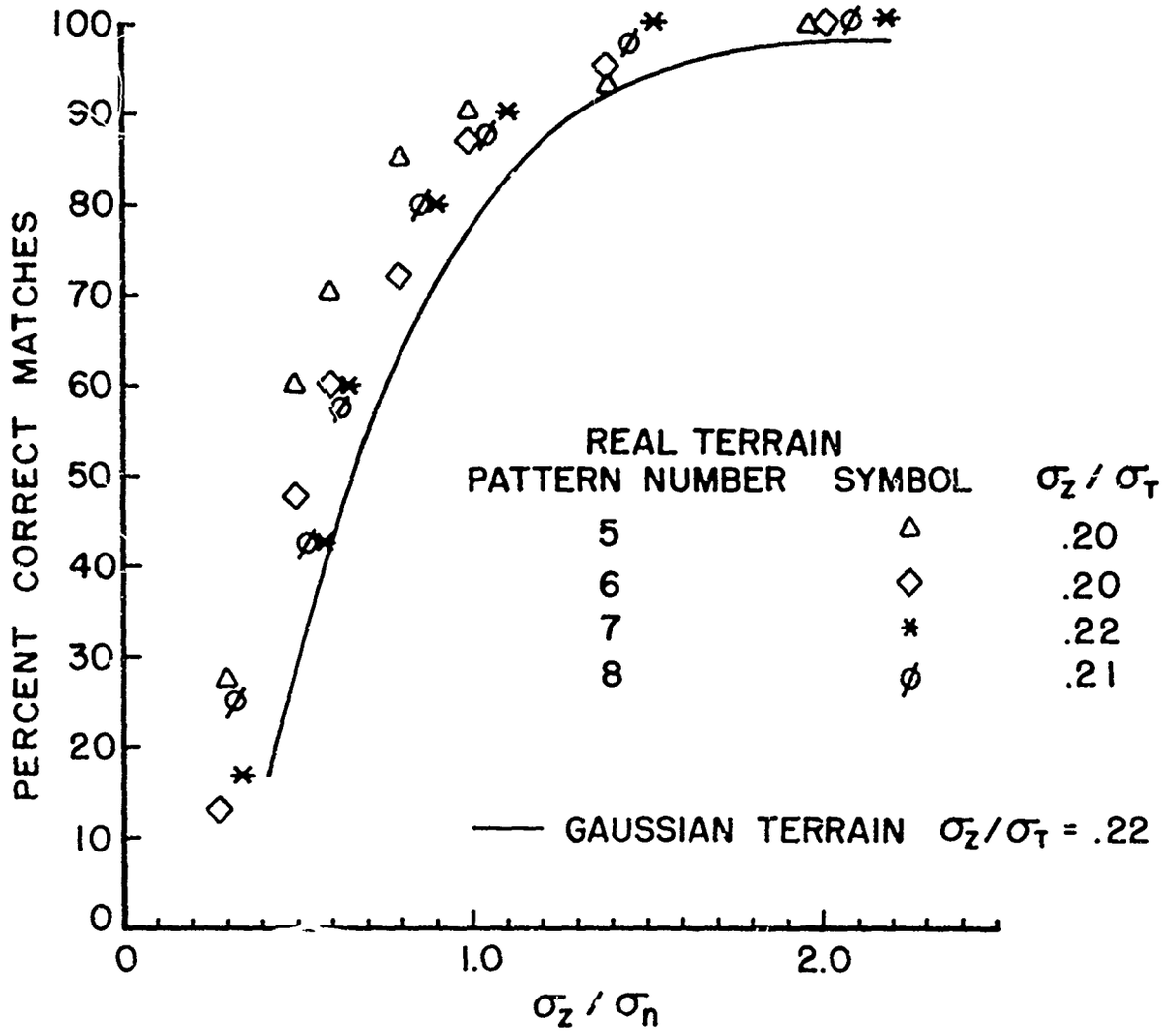


Figure 20. Correct match scores for MSD classifier over Gaussian and real terrain - group 1.

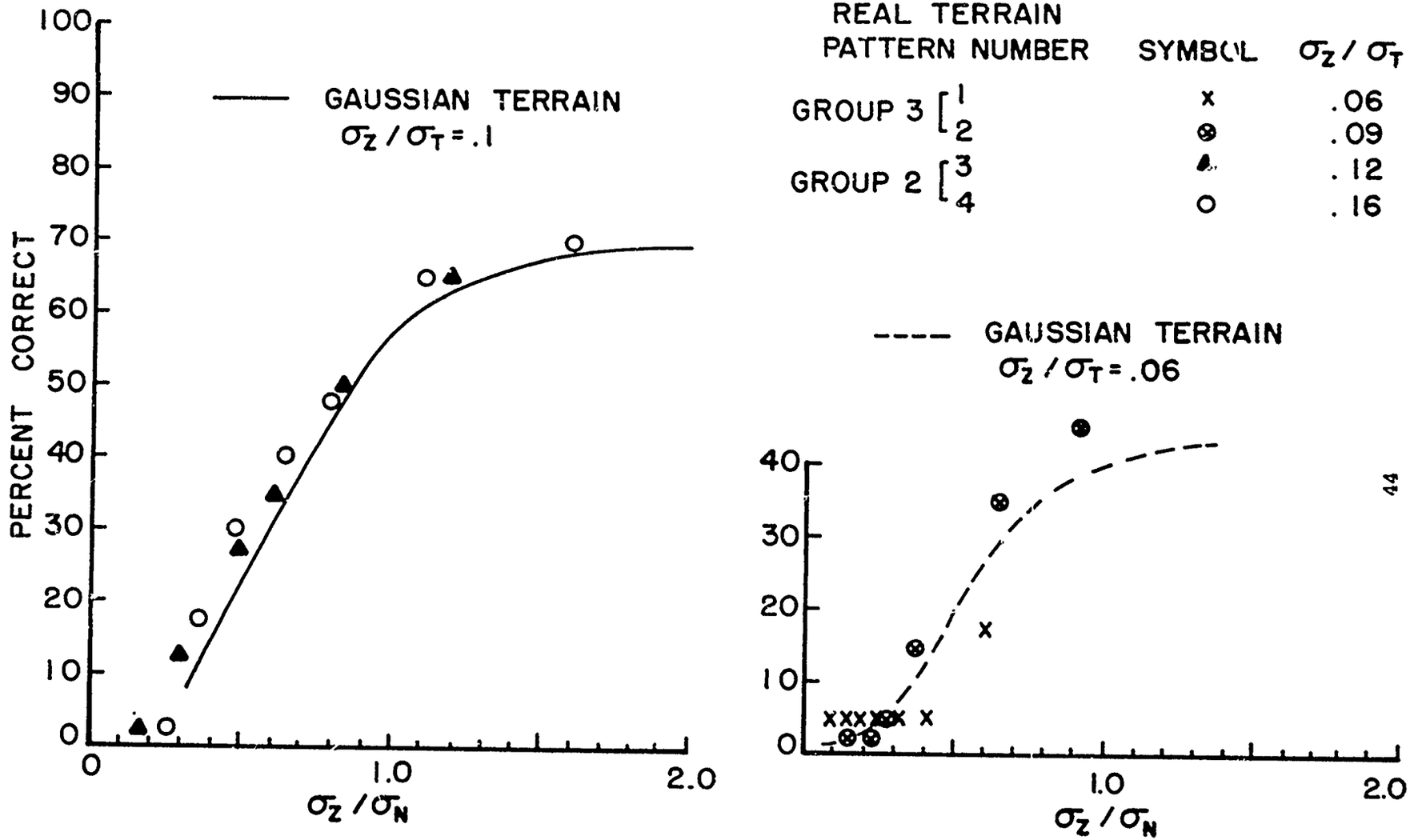


Figure 21. Correct match scores for MSD classifier over Gaussian and real terrain - groups 2 and 3.

DISTRIBUTION OF MISS DISTANCES

Typical miss distance distributions for two real and two Gaussian terrain areas are compared in Figures 22 and 23. The values of σ_z/σ_T for the terrains are comparable in each case. The bin widths in the histograms are 5 cells, corresponding to 2000 ft on the real terrain. MSD and MAD distributions are very close in each case with MSD usually marginally better in the percentage of correct identifications. If 2000-ft errors are acceptable, which is unlikely, the number of correct identifications increase considerably for most signal-to-noise ratios. From all the results, however, completely mountainous terrain with $\sigma_z/\sigma_N \geq 1.4$ gives almost perfect results with MSD or MAD.

CONCLUSIONS AND RECOMMENDATIONS

COMPARISON OF THEORY AND SIMULATION

The results of the theory are compared to the results of the simulation in Figures 24 through 27. The comparison is not direct because the theory predicts how often the false-fix distance will be less than or equal to the terrain correlation length, whereas the simulations count a decision correct only when a correct match is made. To make a direct comparison, we would have to shift the simulation results upwards or the theoretical results downwards. The amount of this shift is not hard to compute - either in theory or in simulation - but it is a computation that we omit in the interest of expediency.

In Figure 24 we see two things: the effect of increasing track length, and the effect of using the present TERCOM mean-removal scheme. In viewing the theoretical predictions, we must recall that the Toeplitz approximation is best for $L_T << d$ or $L_T >> d$ (in the figures, which are computer drawn, N = d, XT = L_T, and XN = L_N). For signal-to-noise ratios greater than about one, the theory and the simulation agree that performance improves with increased track length. In Figure 25 the agreement between theory and simulation is quite good. We again see the beneficial effect of increasing the track length. A comparison of Figures 24 and 25 further

shows that performance will improve slightly in the case where the noise has the same correlation length as the terrain. This same effect is observable in Figure 26, where it is presented explicitly. Again, the agreement between theory and simulation is quite good. Finally, in Figure 27 we see that, according to the simulation, performance decreases monotonically as the terrain correlation length increases (other parameters remaining constant). The theory agrees, except that the drop seems to be too great for intermediate values of L_T: this results in an apparent increase in performance as L_T increases from 4 to, say, 32. The reasons for this apparent discrepancy are believed to be due first to the Toeplitz approximation, and second to the scoring procedure. In this last statement we refer to the previously mentioned, indirect nature of the comparison between theory and simulation. However, it seems entirely appropriate to conclude that performance decreases with increasing terrain correlation length.

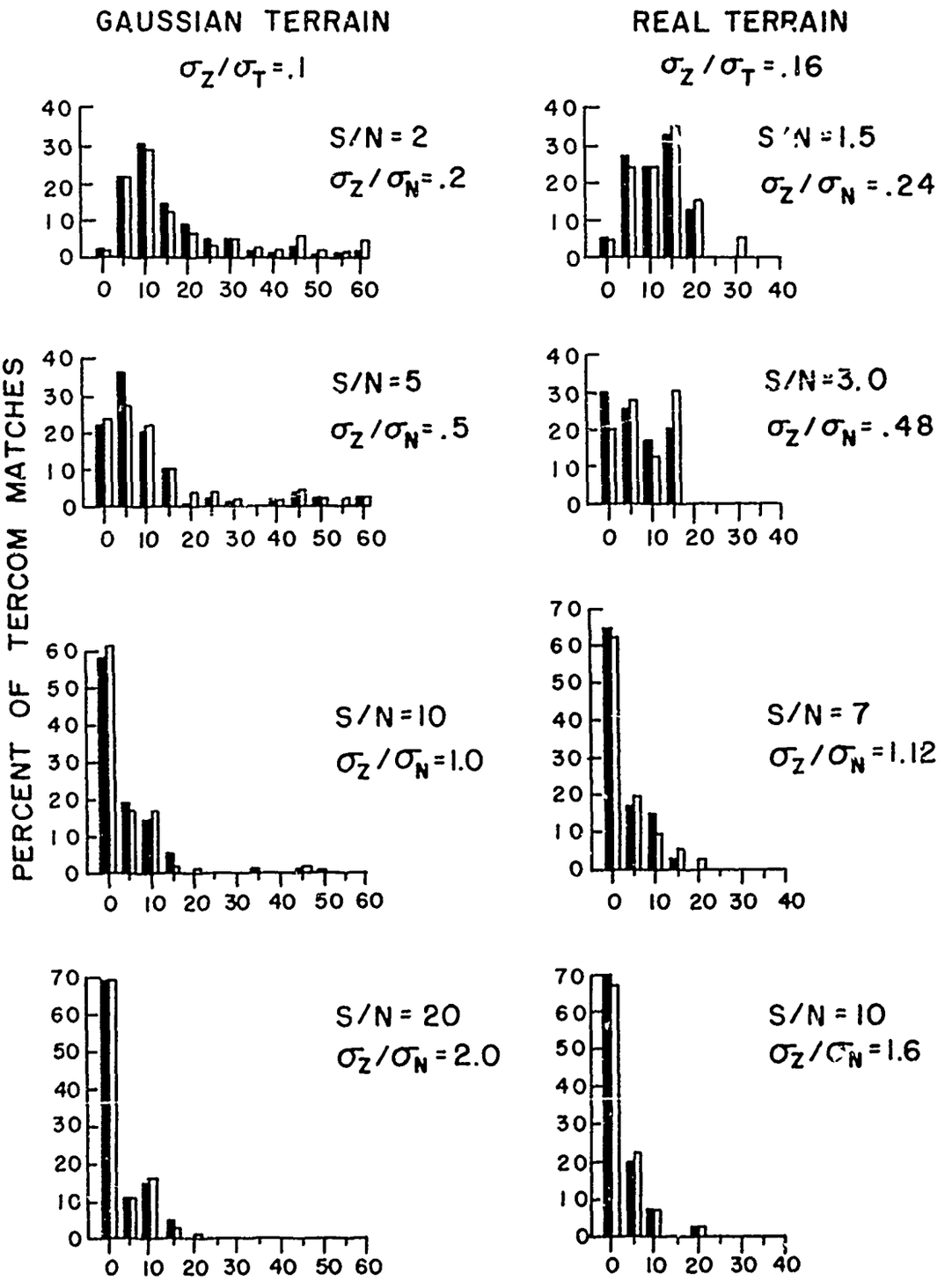


Figure 22. Comparison of TERCOM performance on Gaussian terrain (left column) and real terrain (right column);. Open bars represent the MAD classifier; solid bars represent the MSD classifier; one cell equals 400 feet on the real terrain.

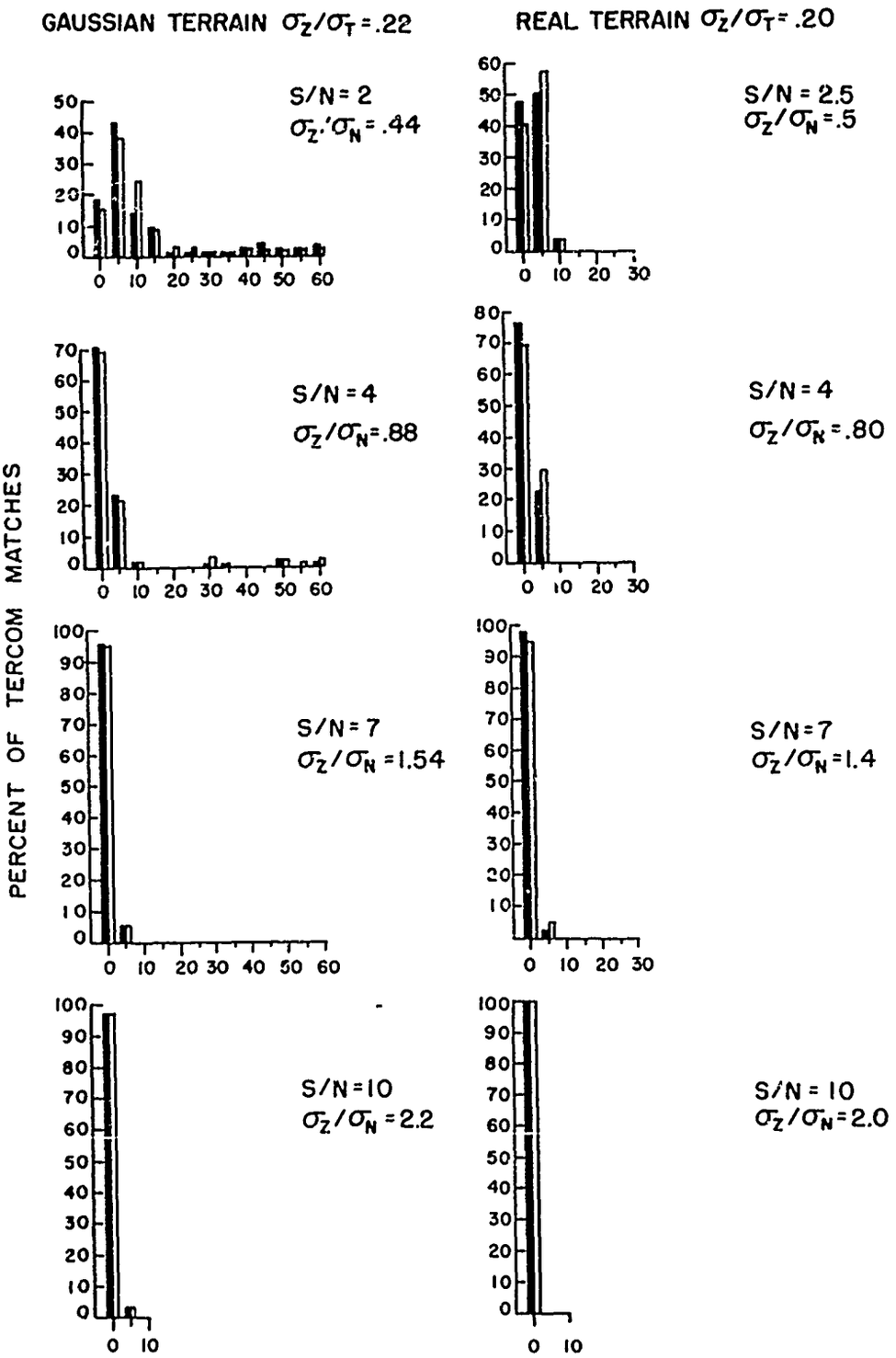


Figure 23. Comparison of TERCOM performance on Gaussian terrain (left column) and real terrain (right column). Open bars represent the MAD classifier; solid bars represent the MSD classifier; on. cell equals 400 feet on the real terrain.

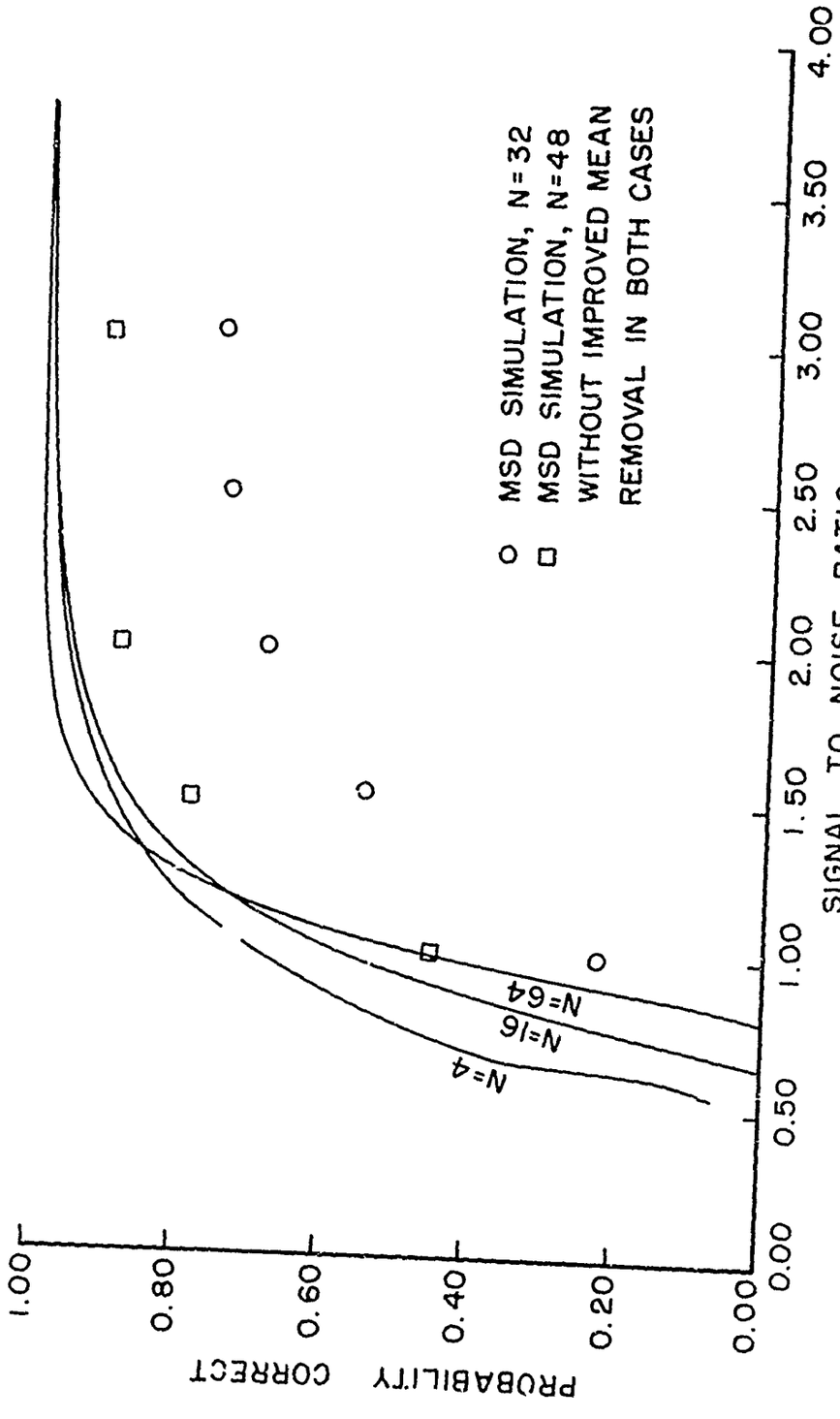


Figure 24. Probability of correct identification as a function of signal-to-noise ratio for three track lengths ($L_T = 4$, $L_N = 0$).

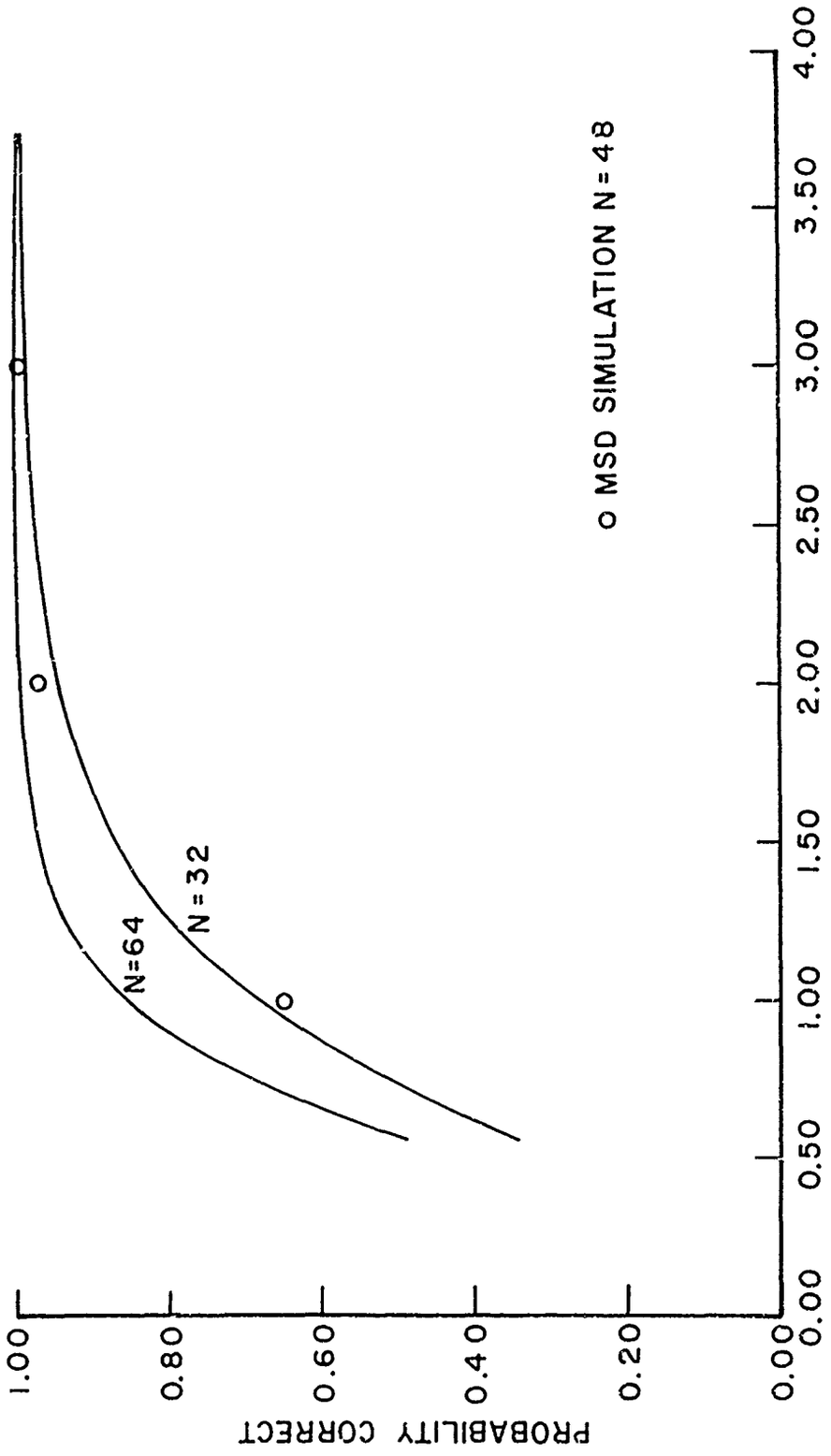


Figure 25. Probability of correct identification as a function of signal-to-noise ratio for two track lengths ($L_T = 4$, $L_N = 4$).

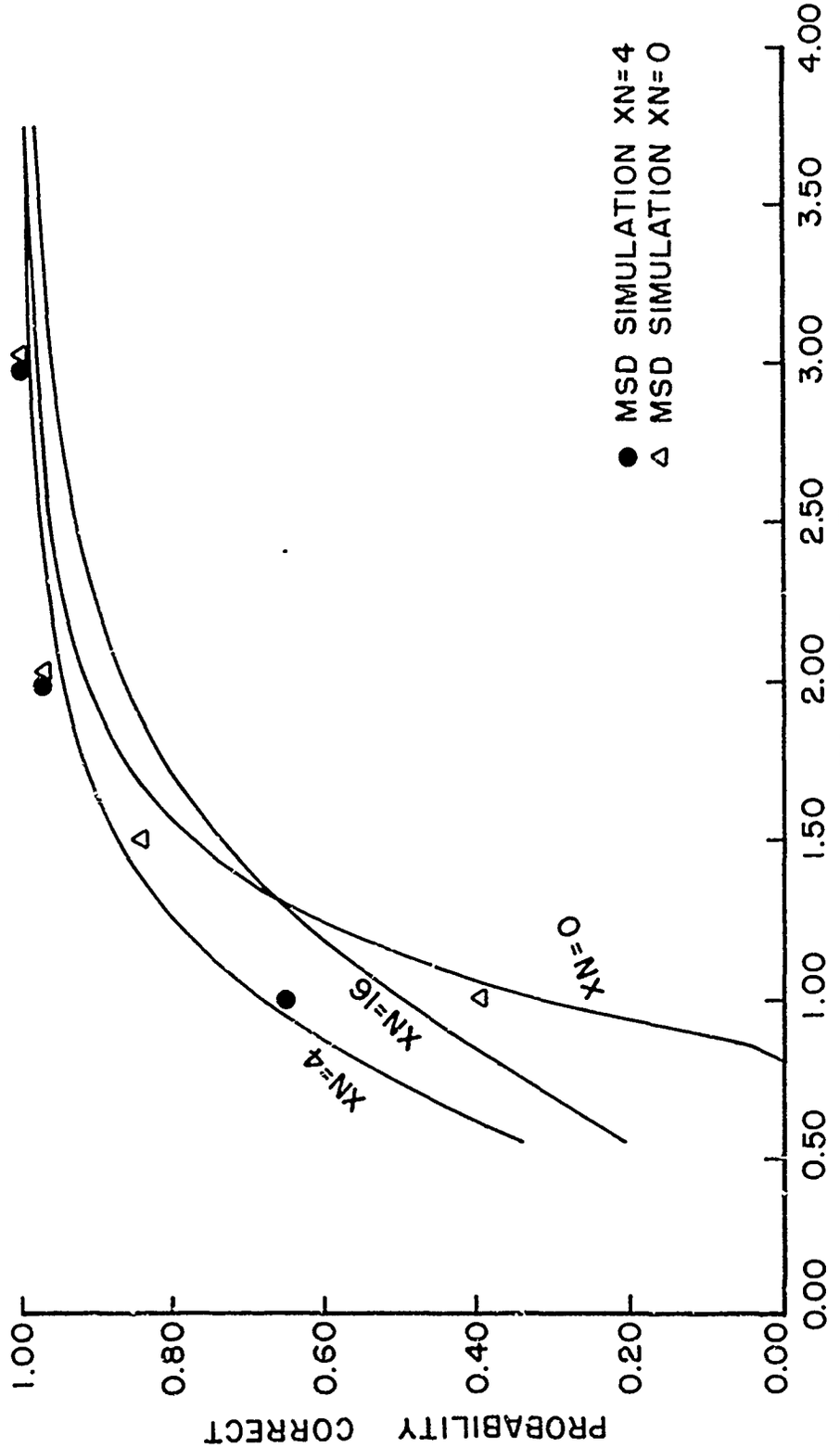


Figure 26. Probability of correct identification as a function of signal-to-noise ratio for three noise correlation lengths (L_T = 4, ℓ = 32).

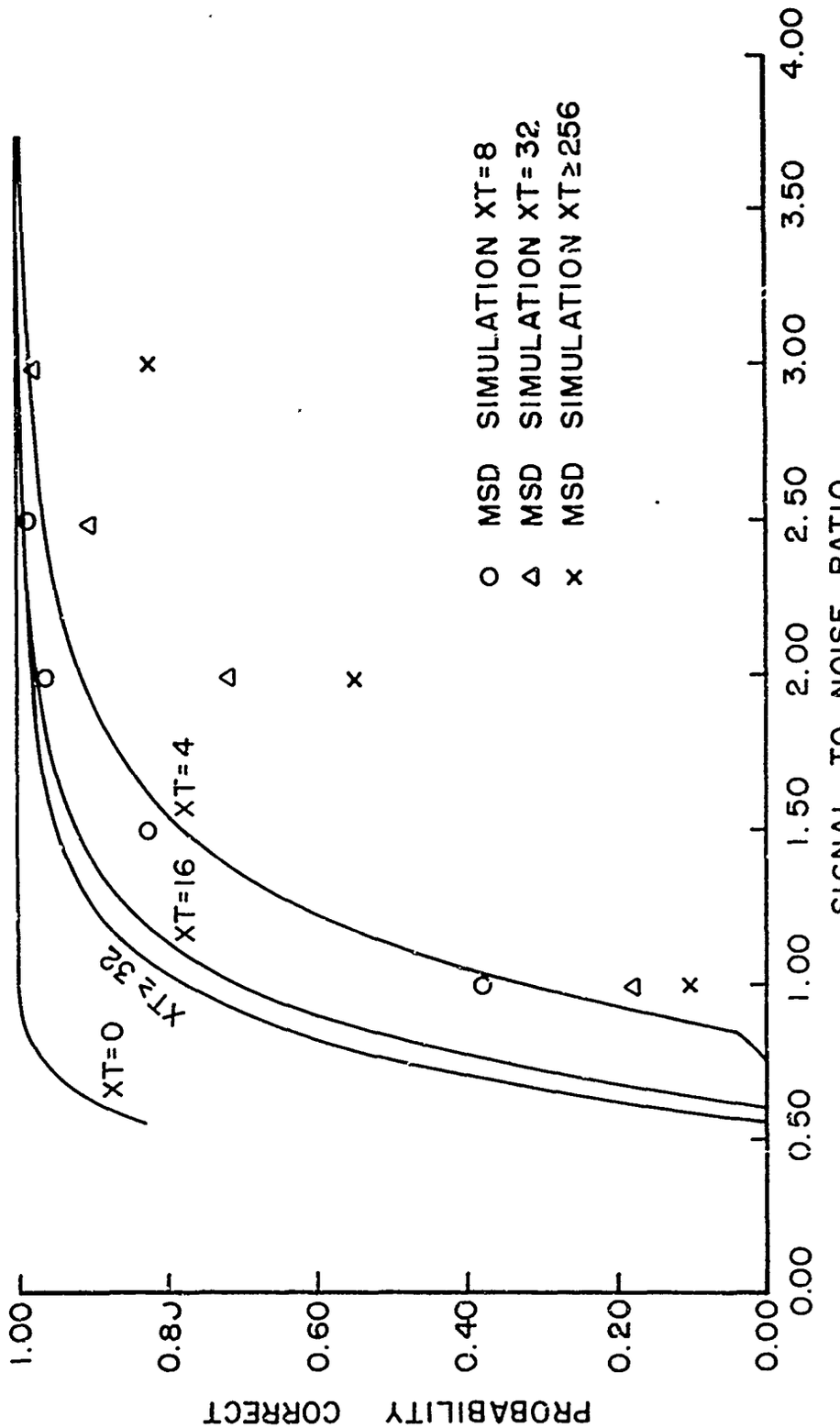


Figure 27. Probability of correct identification as a function of signal-to-noise ratio for four terrain correlation lengths (L_N = 0, d = 32).

AREAS OF FUTURE RESEARCH

The results of this study partially validate a test for TERCOM performance dependent only on three parameters: σ_z and σ_T (which summarize the terrain correlation information) from the terrain and σ_N from the TERCOM system (including all sources of noise). We envision the development, with further research, of a family of performance curves. These would be probability of correct match versus σ_z/σ_N for a family of terrain parameter ratios, σ_z/σ_T. Then we would not be required to simulate a large number of individual flights to determine the suitability of a given terrain area for use as a TERCOM navigational checkpoint. Computation of the required parameter ratios would automatically determine a point on the family of performance curves and thereby produce a value for the expected percentage of correct position identifications.

The parameter ratios σ_z/σ_T and σ_z/σ_N will be useful in analyzing TERCOM performance, but we must now point out some limitations of the present study. The simulations were run, as we mentioned before, with a 48-cell TERCOM track over a 64 x 64 cell terrain grid. This represents the so-called "short track - long matrix" method which assumes that the inertial guidance system has enough accuracy to place the aircraft somewhere over the checkpoint area at the time the scan starts. Different track lengths and different array sizes will change the TERCOM performance curves in a manner predicted by the theory, and indicated by the simulation. Thus the σ_z/σ_N ratio of 1.4, which was quoted as an ideal operating point, would only hold for the specific track and array size considered in the preceding section of the report. We have not considered the "long track - short matrix" method at all in the simulation, although the theory includes this case. Further simulation studies may be desired to determine performance curves for a number of different track lengths and terrain matrix sizes. In particular, model performance curves for the configurations which have been used in flight tests of TERCOM should be established, for it is comparison with actual flight test data that will determine the ultimate usefulness of the Gaussian model and the parameter ratios σ_z/σ_T and σ_z/σ_N. Successful comparison to flight test

results over a wide variety of terrain would show that TERCOM performance characteristics over any terrain area could be predicted by knowing only σ_z and σ_T for the terrain and σ_N for the system noise.

The mention of σ_N brings us to another important limitation of the TERCOM-terrain model presented in this report. We assumed that the system noise was Gaussian. If the real system noise is not well approximated by such a process, flight test performance data could differ from model performance. If the process is non-Gaussian, there might be differences between model and flight test performance even if σ_z/σ_T and σ_z/σ_N are the same for both. We suggest, therefore, a detailed analysis of the pertinent noise sources such as incorrect ground speed, flight path angular deviation, altimeter errors, etc. This would produce greater confidence in the results of this study.

CONCLUDING REMARKS

We have shown that σ_z/σ_T seems to be an adequate description of real terrain, and that Gaussian terrain modified by introducing a certain percentage of flat area closely simulates real terrain.

We have shown that the MSD and MAD classifiers produce almost identical results on both real and Gaussian terrain. This indicates that a theoretical analysis of the more tractable MSD classifier is an adequate theoretical analysis for the MAD classifier.

We have shown agreement between the theory developed herein and the results of the simulation. However, there is a strong assumption in both the theory and the simulation of stationary statistics. This assumption was not valid on the real terrain investigated. The ratio σ_z/σ_T seems to be insensitive to this nonstationarity.

We conclude, on the basis of the theory and the simulation, that TERCOM - modified to include the improved mean-removal scheme - will yield acceptable performance on appropriate terrain. What constitutes "appropriate" can be determined either from σ_T, σ_N, L_T, L_N, and d in the theoretical framework (assuming the stationarity assumption is valid), or from σ_z/σ_T and σ_z/σ_N in the simulation framework. There is no doubt that the quantities σ_z, σ_T, and σ_N are the easiest to obtain.

REFERENCES

1. Summers, T.W. *TERCOM False Fix Study,* ASD-TR-72-74, Vol 4, 1972.

2. Carl, J.W. and G.T. Toussaint. "Historical Note on Minkowski Metric Classifiers," *IEEE Trans. Sys. Man and Cyb.: SMC-1,* No. 4, pp 387-388, October 1971.

3. Moshman, J. "Random Number Generation," *Mathematical Methods for Digital Computers,* Ralston and Wilf, eds., N.Y.: Wiley, 1960.

4. Schwartz, Morton I. "Distribution of the Time-Average Power of a Gaussian Process," *IEEE Trans. Info. Th.: IT-16,* No. 1, pp 17-26, January 1970.

5. Van Trees, H.L. *Detection, Estimation and Modulation Theory, Part I,* N.Y.: Wiley, 1968.

6. Fukunaga, K. *Introduction to Statistical Pattern Recognition,* N.Y.: Academic Press, 1972.

7. Gray, R.M. "On the Assymptotic Eigenvalue Distribution of Toeplitz Matrices," *IEEE Trans. Info. Th.: IT-18,* No. 5, pp 725-730, Nov. 1972.

APPENDIX A

COMPUTER PROGRAMS

1. Program TERCOM results in the performance curves shown in Figures 24 through 27.

2. Program TERC1 generates Gaussian terrain.

3. Program TER is the simulation program.

TERCOM

```
      PROGRAM TERCOM(INPUT, OUTPUT,PLOT)
      DIMENSION EIGT(33), EIGN(33), A(33)
      DIMENSION COV(2,64), NN(1)
      DATA N, XT, XN/32,0., 0./
      CALL PLOT(0.,-1.,-3)
      CALL PLOT(0.,1.,-3)
      CALL SYMBOL(1.,6.,.15,37HPROBABILITY OF CORRECT IDENTIFICATION,0.
     *,37)
      CALL SYMBOL(1.8,5.8,.15,26HAS A FUNCTION OF S/N RATIO,0.,26)
      CALL SYMBOL(3.5,3.5,0.15,5H N = ,0.,5)
      ZN=N
      CALL NUMBER(999.0,999.0,0.15,ZN,0.,1)
      CALL SYMBOL(3.5,3.3,0.15,5HXN = ,0.,5)
      CALL NUMBER(999.0,999.0,0.15,XN,0.,1)
      CALL SYMBOL(3.5,3.1,0.15,20HXT IS PARAMETER WITH,0.,20)
      CALL SYMBOL(3.5,2.90,0.15,21HVALUES: 0,4,16       ,0.,21)
      CALL AXIS(0.,0.,21HSIGNAL TO NOISE RATIO,-21,8.,0.,0.,.5)
      CALL AXIS(0.,0.,19HPROBABILITY CORRECT,19,5.,90.,0.,.2)
  200 FORMAT(5X,F6.2,5X,F12.6,5X,F9.6,5X,F9.6,10X,I2,5X,F9.6,5X,I2,5X,F9
     1.6)
  201 FORMAT(38H1PROBABILITY(CORRECT) FOR THE CASE N = ,I3,6H XT = ,F7.3
     1,10H AND XN = ,F7.3//)
  202 FORMAT(7X,3HSNR,11X,1HR,13X,4HPTAU,8X,10HP(CORRECT),9X,2HNN,10X,2H
     1CN,7X,2HNT,10X,2HCT//)
      DO 2000 NRUN=1,3
      SNR = 0.55
      NHALF = N/2 + 1
      NN(1) = N
      N1 = N + 1
      N2 = NHALF + 1
      IF(XT.EQ.0.)GO TO 2
      DO 11 I=1,NHALF
      COV(1,I) = EXP(FLOAT(1-I)/XT)
   11 COV(2,I) = 0.
       DO 12 I=N2,N
      COV(1,I) = EXP(FLOAT(I-N1)/XT)
   12 COV(2,I) = 0.
      CALL FOURT(COV,NN,1,-1,0,0)
      DO 7 I=1,NHALF
    7 EIGT(I) = COV(1,I)
      GO TO 14
    2 CONTINUE
      DO 13 I=1,NHALF
   13 EIGT(I) = 1.
   14 CONTINUE
      IF(XN.EQ.0.)GO TO 4
      DO 8 I=1,NHALF
      COV(1,I) = EXP(FLOAT(1-I)/XN)
    8 COV(2,I) = 0.
      DO 9 I=N2,N
      COV(1,I) = EXP(FLOAT(I-N1)/XN)
    9 COV(2,I) = 0.
      CALL FOURT(COV,NN,1,-1,0,0)
      DO 10 I=1,NHALF
   10 EIGN(I) = COV(1,I)
```

Initialize program: set-up for 64×64 covariance matrix (max.). Track length (N) = 32; $L_T(XT)=0.$ initially; $L_N(XN)=0.$

Initialize parameters. Obtain 3 runs with XT (terrain correlation) as the independent variable

Check for 0 correlation length.

Build one row of terrain correlation matrix according to Gauss-Markov statistic and Toeplitz approximation.

Compute Eigenvalues (FFT).

Store Terrain Eigenvalues

Special case of XT=0.

Check for 0 correlation length.

Build one row of noise correlation matrix according to Gauss-Markov statistic and Toeplitz approximation.

Compute eigenvalues (FFT).

TERCOM

```
      GO TO 15
    4 CONTINUE
      DO 15 I=1,NHALF
   15 EIGN(I) = 1.
   15 CONTINUE
      CALL APROX(NHALF,NN,CN,EIGN)
      PRINT 201, N, XT, XN
      PRINT 202
      DO 1000 NCTR=1,65
      VR = 2.*SNR*SNR
      DO 5 I=1,NHALF
    5 A(I) = VR*EIGT(I) + EIGN(I)
      CALL APROX(NHALF,NT,CT,A)
      R = CT/CN
      ERTAU = RTAU(NN,NT,R)
      IF(ERTAU.GT.0.5)ERTAU = 0.5
      B = 1. - 2.*ERTAU
      PRINT 200, SNR, R, ERTAU, B, NN, CN, NT, CT
      PSNR=SNR*2. ; PB=5.*B
      IF(NCTR.EQ.4.OR.NCTR.EQ.6) GO TO 500
      IF(NCTR.EQ.1) CALL PLOT(PSNR,PB,3)
      CALL PLOT(PSNR,PB,2)
  500 CONTINUE
      SNR=SNR+0.05
 1000 CONTINUE
      IF(NRUN.EQ.1) XT=1.
 2001 XT=4*XT
      CALL PLOTE
      STOP
      END
```

} Store Noise Eigenvalues
} Special case of XN=0.

} Compute pulse-approximation for noise process.

} Compute pulse-approximation for terrain + noise process
SNR = σ_T/σ_N

} Compute $P(m<m_0)$.

} Compute total probability (correct).

} Increment SNR for purpose of plot.

} Change parameter XT; result is 3 plots parameterized by XT, each a function of SNR having N=32 and XN=C.

APROX

```
      SUBROUTINE APROX(NHALF,K,C,A)
      DIMENSION A(NHALF)
      C = A(1)
      EMAX = A(1)
      K = 1
      DO 3 I=2,NHALF
      SUM = 0.
      DO 1 J=1,I
    1 SUM = SUM + A(J)
      E = SUM**2/FLOAT(I)
      IF(E.LT.EMAX)GO TO2
      EMAX = E
      K = I
      C = SUM/FLOAT(I)
    2 CONTINUE
    3 CONTINUE
      RETURN
      END
```

RTAU

```
      FUNCTION RTAU(NN,NT,R)
      R1 = R+1.
      C = (R/R1)**NN
      SUM = 1.
      IF(NT.EQ.1)GO TO 2
      A = 1
      NT1 = NT-1
      XNN = NN-1
      DO 1 I=1,NT1
      XI = I
      A = A*(XNN+XI)/(XI*R1)
    1 SUM = SUM + A
    2 RTAU = 1. - C*SUM
      RETURN
      END
```

TERC1

```
      PROGRAM TERC1(INPUT,OUTPUT,TAPE1)
      DIMENSION A(2,64,64),NN(2)
      DATA NN,TC,P,B/64,64,4.,5.,1./
      CALL RANSET(P)
      DO 1000 NCTR=1,10
      DO 10 I=1,2
      DO 10 J=1,64
      DO 10 K=1,64
      A(I,J,K)=0.
   10 CONTINUE
      DO 20 J=1,33
      DO 20 K=1,33
      CJ1=FLOAT(J)
      CK1=FLOAT(K)
      C=SQRT((CJ1-1.)**2+(CK1-1.)**2)
      A(1,J,K)=B*EXP(-C/TC)
   20 CONTINUE
      DO 30 K=1,33
      DO 30 J=2,32
      J1=66-J
      A(1,J1,K)=A(1,J,K)
   30 CONTINUE
      DO 40 J=1,64
      DO 40 K=2,32
      K1=66-K
      A(1,J,K1)=A(1,J,K)
   40 CONTINUE
      CALL FOURT(A,NN,2,-1,0,0)
      DO 60 J=1,64
      DO 60 K=1,33
      F2=A(1,J,K)
      CALL RANDOM(F2,RE,XI)
      A(1,J,K)=RE
      A(2,J,K)=XI
   60 CONTINUE
      DO 70 J=34,64
      J1=66-J
      A(1,J,1)=A(1,J1,1)
      A(2,J,1)=-A(2,J1,1)
      A(1,J,33)=A(1,J1,33)
      A(2,J,33)=-A(2,J1,33)
   70 CONTINUE
      DO 80 K=34,64
      K1=66-K
      A(1,1,K)=A(1,1,K1)
      A(2,1,K)=-A(2,1,K1)
      A(1,33,K)=A(1,33,K1)
      A(2,33,K)=-A(2,33,K1)
   80 CONTINUE
      DO 85 J=2,32
      DO 85 K=2,32
      J1=66-J
      K1=66-K
      A(1,J1,K1)=A(1,J,K)
      A(2,J1,K1)=-A(2,J,K)
```

TERC1

```
 85 CONTINUE
    DO 90 J=2,32
    DO 90 K=2,32
    K1=66-K
    J1=66-J
    A(1,J,K1)=A(1,J1,K)
    A(2,J,K1)=-A(2,J1,K)
 90 CONTINUE
    CALL FOURT(A,NN,2,1,0,0)
    CALL RANGET(P)
1000 WRITE(1) NCTR, TC, P, B, ((A(1,I,J), I=1,64), J=1,64)
    PRINT 102, NCTR, P
102 FORMAT(10X,*NCTR=*I2,5X,*P=*E15.7)
    REWIND 1
    STOP
    END
```

RANDOM

```
      SUBROUTINE RANDOM(F2,RE,XI)
      IF(F2.GT.0.)GO TO 1
      RE=0.
      XI=0.
      GO TO 3
    1 F=SQRT(F2)
      ZMU=0.
      SIGMA=F/3.
    2 X=RANF(DUM)
      Y=RANF(DUM)
      CC=SQRT(-2.*ALOG(X))*SIGMA
      RE=CC*COS(6.28313*Y)+ZMU
      IF(ABS(RE).GT.F)GO TO 2
      XI=SQRT(F2-RE**2)
      IF(X.GT..5)XI=-XI
    3 RETURN
      END
```

TER

```
      PROGRAM TER (INPUT,OUTPUT,TAPE1)
      DIMENSION A(64,64),S(64),AB(64,64)
      DIMENSION RN(64)
      RO=0.          ------   NOISE CORRELATION COEFFICIENT
      E=.7642672
      NL=48          ------   SET TRACK LENGTH
      NBL=64         ------   SET ARRAY SIZE
      NL5=NBL-NL
      BL5=FLOAT(NL5)
      CALL RANSET(E)
      PRINT 107,E
      PRINT 110
      ANUM=SNUM=DMSD=DMAD=0.
      DO 70 NC=1,10
      READ(1)NCTR,TC,P,B,A -  READ TERRAIN PATTERN
      DO 5 I=1,64
      DO 5 J=1,2
      A(I,J)=0.
    5 CONTINUE
      PRINT 104,NC
      SUM10=0.
      SUM11=0.
      RA=2.  --  SET SIGNAL TO NOISE RATIO
      FMEAN=0.
      IK=JK=64
      BN=FLOAT(IK*JK)
      SUM=0.
      BN2=FLOAT(IK-1)
      DO 10 I=1,IK
      DO 10 J=1,JK
      SUM=SUM+A(I,J)
   10 CONTINUE
      AVE=SUM/BN
      SUM=0.
      DO 20 J=1,JK
      DO 20 I=1,IK
      SUM=SUM+(A(I,J)-AVE)**2
   20 CONTINUE
      SIGT=SQRT(SUM/BN)    -σT COMPUTATION
      SIGNO=SIGT/RA        σN COMPUTATION
      ID=1
      IF(RO.NE.0.)GO TO 22
      RO1=0.
      GO TO 25
   22 IF(RO.NE.1.)GO TO 23
      RO1=1.
      GO TO 25
   23 SCN=-1./ALOG(RO)    - CORRELATION LENGTH
      RO1=EXP(-SQRT(2.)/SCN) - CORRELATION BETWEEN
   25 ACA=RO/(1.+RO1)
      ACB=SQRT(1.-2.*((RO)**2)/(RO1+1.))
      CALL RAND2(RN1,RN2,SIGNC,FMEAN,ID)
      RN3=RN2
      DO 30 I=1,64,2
      CALL RAND2(RN1,RN2,SIGNC,FMEAN,ID)
```

```
TER
      RN(I)=RO*RN3+SQRT(1.-RO**2)*RN1                    | NOISE
      RN(I+1)=RO*RN(I)+SQRT(1.-RO**2)*RN2                | PATTERN
      SUM10=RN(I)**2+RN(I+1)**2+SUM10                    | COMPUTATION
      SUM11=RN(I)+RN(I+1)+SUM11
      AB(I,1)=A(I,1)+RN(I)
      AB(I+1,1)=A(I+1,1)+RN(I+1)
      RN3=RN(I+1)
   30 CONTINUE
      DO 31 J=2,64
      CALL RAND2(RN1,RN2,SIGNO,FMEAN,ID)
      RN(1)=RO*RN(1)+SQRT(1.-RO**2)*RN1
      RN(2)=ACA*(RN(1)+RN(2))+ACB*RN2                    | ADD
      SUM10=RN(1)**2+RN(2)**2+SUM10                      | NOISE
      SUM11=RN(1)+RN(2)+SUM11                            | TO
      AB(1,J)=A(1,J)+RN(1)                               | TERRAIN
      AB(2,J)=A(2,J)+RN(2)
      DO 31 I=3,64,2
      CALL RAND2(RN1,RN2,SIGNO,FMEAN,ID)
      RN(I)=ACA*(RN(I-1)+RN(I))+ACB*RN1
      RN(I+1)=ACA*(RN(I)+RN(I+1))+ACB*RN2
      SUM10=RN(I)**2+RN(I+1)**2+SUM10
      SUM11=RN(I)+RN(I+1)+SUM11
      AB(I,J)=A(I,J)+RN(I)
      AB(I+1,J)=A(I+1,J)+RN(I+1)
   31 CONTINUE
      SUM=0.
      DO 52 J=1,JK
      DO 52 I=1,IK
      SUM=SUM+AB(I,J)
   52 CONTINUE
      SUM=SUM/4096.
      DO 53 J=1,JK
      DO 53 I=1,IK
      AB(I,J)=AB(I,J)-SUM
   53 CONTINUE
      SDEVN=SQRT(SUM10/4096.-(SUM11/4096.)**2)
      IF(SDEVN.NE.0.)GO TO 40
      SNR=11.11
      GO TO 41
   40 SNR=SIGT/SDEVN    - COMPUTE ACTUAL S/N
   41 PRINT113,SDEVN,SIGT,SNR
      ID=2
      DO 70 L=1,10
      CALL RAND2(RN1,RN2,SIGNO,FMEAN,ID)
      KI1=IFIX(81.5*RN1)+1
      KI2=KI1+NL-1
      KJ1=IFIX(63.*RN2)+1                    COMP. E TRAY
      DO 51 I=KI1,KI2                        STARTING COORD.
      M=I-KI1+1
      S(M)=A(I,KJ1)
   51 CONTINUE
      SUM=0.
      DO 55 I=1,NL
      SUM=SUM+S(I)
   55 CONTINUE
```

```
TER

      SUM=SUM/NL
      DO 56 I=1,NL
      S(I)=S(I)-SUM
   56 CONTINUE
      NL2=64-NL+1
      SUMR=32000.
      SUMR2=1.E7
      DO 62 J=1,JK
      DO 62 K=1,NL2
      SUM=0.
      SUM2=0.
      SUM5=0.
      DO 58 I=1,NL                    COMPUTE
      II=I+K-1                      MSD AND MAD
      SUM5=AB(II,J)+SUM5              RESULTS
   58 CONTINUE
      SUM5=SUM5/FLOAT(NL)
      DO 60 I=1,NL
      II=I+K-1
      SUM=SUM+ABS(S(I)-AB(II,J)+SUM5)
      SUM2=SUM2+(S(I)-AB(II,J)+SUM5)**2
   60 CONTINUE
      IF(SUM.GE.SUMR)GO TO 61
      SUMR=SUM
      KIR=K
      KJR=J
   61 IF(SUM2.GE.SUMR2)GO TO 62
      SUMR2=SUM2
      KIR2=K
      KJR2=J
   62 CONTINUE
      DMAD=SQRT((FLOAT(KI1-KIR))**2+(FLOAT(KJ1-KJR))**2)+DMAD
      DMSD=SQRT((FLOAT(KI1-KIR2))**2+(FLOAT(KJ1-KJR2))**2)+DMSD
      IF(KI1.EQ.KIR.AND.KJ1.EQ.KJR)GO TO 63
      IF(KI1.EQ.KIR2.AND.KJ1.EQ.KJR2)GO TO 66   COMPUTE ERROR
      GO TO 67                                    M
   63 IF(KI1.EQ.KIR2.AND.KJ1.EQ.KJR2)GO TO 65
      PRINT 105,KIR,KJR,KI1,KJ1,KIR2,KJR2
      SNUM=SNUM+1.
      GO TO 70
   65 PRINT 106,KIR,KJR,KIR2,KJR2
      GO TO 70
   66 PRINT 108,KI1,KJ1,KIR,KJR,KI1.KJ1
      ANUM=ANUM+1.
      GO TO 70
   67 PRINT 109,KI1,KJ1,KIR,KJR,KI1,KJ1,KIR2,KJR2
      ANUM=ANUM+1.
      SNUM=SNUM+1.
   70 CONTINUE
      DMAD=DMAD/ANUM
      DMSD=DMSD/SNUM
      REWIND 1
      PRINT 103,RA
      PRINT 111,DMAD,DMSD
      PRINT 112,RO
```

```
TER

      PRINT 114,NL
      CALL RANGET(DD)
       PRINT 107,DD
      STOP
  103 FORMAT(1X,*SIGMA T / SIGMA N =*,F4.1)
  104 FORMAT(1X,*NCTR=*,I3)
  105 FORMAT(1X,*CORRECT I.D.AT*,2I3,10X,2I3,1X,*MISTAKEN FOR*,2I3)
  106 FORMAT(1X,*CORRECT I.D.AT*,2I3,10X,*CORRECT I.D.AT*,2I3)
  107 FORMAT(1X,*RANDOM NUMBER=*,E15.7)
  108 FORMAT(1X,2I3,1X,*MISTAKEN FOR*,2I3,5X,*COPRECT I.D.AT*,2I3)
  109 FORMAT(1X,2I3,1X,*MISTAKEN FOR*,2I3,5X,2I3,1X,*MISTAKEN FOR*,2I3)
  110 FORMAT(7X,*MAD*,32X,*MSD*)
  111 FORMAT(1X,*AVE. MAD ERROR=*,1X,F5.1,11X,*AVE. MSD ERROR=*,1X,F5.1)
  112 FORMAT(1X,*RO=*,F5.2)
  113 FORMAT(60X,*NOISE SIGMA=*,E15.7,1X,*SIGNAL SIGMA=*,E15.7,
     1*S/N=*,F5.2)
  114 FORMAT(1X,*TRACK LENGTH =*,I3)
      END
```

```
      SUBROUTINE RAND2(RN1,RN2,SIGNO,FMEAN,ID)
      X=RANF(DUM)
      Y=RANF(DUM)
      IF(ID.EQ.1)GO TO 5
      RN1=X
      RN2=Y
      GO TO 10
    5 CC=SQRT(-2.*ALOG(X))*SIGNO
      RN1=CC*COS(6.28313*Y)+FMEAN
      RN2=CC*SIN(6.28313*Y)+FMEAN
   10 RETURN
      END
```

ID=1 FOR GAUSSIAN DISTRIBUTION
ID≠1 FOR UNIFORM DISTRIBUTION

APPENDIX B

METHOD FOR COMPUTING VALUES FOR GAUSSIAN CORRELATED NOISE IN ONE DIMENSION

Let R_n be a Gaussian random variable of mean zero. We wish to generate a string of numbers S_n which are Gaussian with mean zero and any given correlation coefficient ρ.

We choose $$S_n = w_1 S_{n-1} + w_2 R_n \ , \qquad \text{(B1)}$$

where w_1 and w_2 are weights to be determined.

Now, take the variance of S_n

$$\text{Var}\ (S_n) = w_1^2\ \text{Var}\ (S_{n-1}) + w_2^2\ \text{Var}\ (R_n)$$

or more simply $$\sigma^2 = w_1^2\sigma^2 + w_2^2\sigma_N^2$$

Furthermore, we chose $\sigma = \sigma_N$ to simplify computation

which gives us $$1 = w_1^2 + w_2^2$$

Now multiply Eq B1 by S_{n-1} and take the expectation of both sides

$$E(S_n S_{n-1}) = Ew_1(S_{n-1})^2 + Ew_2(S_{n-1}R_n)$$

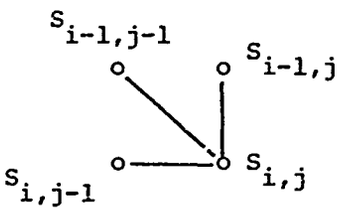
$$\sigma^2\rho = w_1\sigma^2 + 0$$

$$\rho = w_1$$

$$1 = \rho^2 + w_2^2 \quad \text{or} \quad w_2 = \sqrt{1 - \rho^2}$$

thus $$S_n = \rho S_{n-1} + \sqrt{1 - \rho^2}R_n$$

This can be extended to two dimensions by making $S_{i,j}$ depend on $S_{i-1,j}$, $S_{i-1,j-1}$ and $S_{i,j-1}$



we have then

$$E(R_n) = \sigma^2$$

$$E(S_{i,j}S_{i,j}) = \sigma^2$$

$$E(S_{i,j}S_{i-1,j}) = E(S_{i,j}S_{i,j-1}) = \rho\sigma^2$$

$$E(S_{i,j}S_{i-1,j-1}) = \rho_1\sigma^2$$

Let $S_{i,j} = w_1S_{i-1,j} + w_2S_{i,j-1} + w_3R_n$ (B2)

where again $0 \leq w_1 \leq 1$, $0 \leq w_2 \leq 1$, $0 \leq w_3 \leq 1$

and $\text{Var}\,(S_{i,j}) = \text{Var}\,R_n$

$$\text{Var}\,S_{ij} = E(w_1S_{i-1,j} + w_2S_{i,j-1} + w_3R_n)^2$$

$$\sigma^2 = w_1^2\sigma^2 + w_2^2\sigma^2 + w_3^2\sigma^2 + 2w_1w_2\rho_1\sigma^2$$

or $1 = w_1^2 + w_2^2 + w_3^2 + 2w_1w_2\rho_1$ (B3)

Multiply Eq B2 by $S_{i-1,j}$ and take the expectation of both sides. This gives:

$$\rho\sigma^2 = w_1\sigma^2 + w_2\rho_1\sigma^2 + 0$$

$$\rho = w_1 + w_2\rho_1 \quad \text{(B4)}$$

Multiply Eq B2 by $S_{i,j-1}$ and take the expectation of both sides to get

$$\rho\sigma^2 = w_1\rho_1 + w_2\sigma^2 + 0$$

$$\rho = w_1\rho_1 + w_2 \quad \text{(B5)}$$

Now multiply Eq B4 by ρ_1 and subtract Eq B5 from the result to get

$$\rho\rho_1 = \rho_1w_1 + w_2\rho_1^2$$

$$-\rho = -\rho_1w_1 - w_2$$

$$\rho(\rho_1-1) = w_2(\rho_1^2-1)$$

$$w_2 = \frac{\rho(\rho_1-1)}{(\rho_1^2-1)} = \frac{\rho}{\rho_1+1}$$

Similarly we find that $w_1 = w_2 = \frac{\rho}{\rho_1+1}$

Now substituting for w_1 and w_2 in Eq B3, we get

$$1 = \frac{2\rho^2}{(\rho_1+1)^2} + \frac{2\rho_1\rho^2}{(\rho_1+1)^2} + w_3^2$$

or $$w_3 = \sqrt{1 - \frac{(2\rho^2 + 2\rho_1\rho^2)}{(\rho_1 + 1)^2}} = \sqrt{1 - \frac{2\rho^2}{(\rho_1 + 1)}}$$

Thus $$S_{i,j} = \frac{\rho}{\rho_1 + 1} S_{i-1,j} + \frac{\rho}{\rho_1 + 1} S_{i,j-1} + \sqrt{1 - \frac{2\rho^2}{\rho_1 + 1}} R_n$$

When $\rho = 0$, $S_{i,j} = R_n$ and when $\rho = \rho_1 = 1$, $S_{ij} = \frac{1}{2} S_{i-1,j} + \frac{1}{2} S_{i,j-1}$.

There are two ways to specify the characteristics of the correlated noise we wish to generate. The noise may be characterized by a correlation length τ or a correlation coefficient ρ. In both cases an exponential autocorrelation function is assumed for the noise.

If the $S_{i,j}$ are treated as samples from a continuous function, we can determine ρ and ρ_1 when the correlation length τ is given. If the correlation length is τ cells, ρ is the value of the correlation function one cell away from the origin. Thus we have

$$\rho = e^{-\frac{1}{\tau}}$$

Since S_{i-1}, S_{j-1} is 1.414 cells distant from $S_{i,j}$,

$$\rho_1 = e^{-\frac{1.414}{\tau}}$$

If the noise is specified in terms of a correlation coefficient ρ, we can determine τ from the relation $\tau = 1/-\ln\rho$. Then, ρ_1 can be determined as shown above.

U.S.Government Printing Office: 1974 — 758-436/697